19970205 033

A PRELIMINARY DESIGN OF A STANDARDIZED SPACECRAFT BUS FOR SMALL TACTICAL SATELLITES

THESIS

Written by GSO & GSE team

AFIT/GSE/GSO/ENY/96D-1

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

A PRELIMINARY DESIGN OF A
STANDARDIZED SPACECRAFT BUS FOR
SMALL TACTICAL SATELLITES

THESIS

Written by GSO & GSE team

AFIT/GSE/GSO/ENY/96D-1

Approved for public release; distribution unlimited

Disclaimer

The views expressed in this thesis are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

AFIT/GSE/GSO/ENY/96D-1

A PRELIMINARY DESIGN OF A STANDARDIZED SPACECRAFT BUS FOR SMALL TACTICAL SATELLITES

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

in Partial Fulfillment of the Requirements for the

Degree of Master of Science

Gerald Ashby, B.S.
Capt, USAF

Darren Buck, B.S.
Capt, USAF

Robert Carneal IV, B.S.
1Lt, USAF

Tansel Cokuysal, B.S.
1Lt, Turkish AF

Ahmet Donmez, B.S.
1Lt, Turkish AF

James From, B.S., M.A.
Capt, USAF

Todd Krueger, B.S.
Capt, USAF

Brian Robinson, B.S.
1Lt, USAF

November 1996

Approved for public release; distribution unlimited

Preface

The following document is the culmination of the work performed by a team of eight graduate engineering students assigned to the Air Force Institute of Technology (AFIT). The students compiled this document while performing a systems engineering design study to create a small standardized tactical satellite bus for the Phillips Laboratory. This document is divided into three separate volumes. Each volume is an integrated element of the student thesis but it can also serve as a stand alone document.

The first volume is the Executive Summary. The purpose of the Executive Summary is to present a synopsis of the design study results to the sponsor at the Phillips Laboratory. This volume includes information on the methods employed during the study, the scope of the problem, the value system used to evaluate alternatives, tradeoff studies performed, modeling tools utilized to create and analyze design alternatives, recommendations and implications of the alternatives, and areas where future research should be considered.

The second volume is a detailed account of the design process. The steps of the team's innovative design process and the team organization are initially presented. Each phase of the design study is discussed in subsequent sections. Phase I provides accounts of the team's initial attempt to apply a well known systematic approach to satellite design. Efforts concentrate on defining the problem posed by the sponsor. "First cuts" at developing analysis tools and models are performed. Additionally, different alternatives are generated as possible solutions to the problem. An initial analysis and evaluation is

performed to define an initial solution space, and to verify the analysis tool. Phase II is an iterative step in the design process and serves as a reservoir for the team's most meaningful work. The team realized that a new systematic approach had to be applied to the study. This phase provides the results of the application of that innovative approach. It is here that the understanding of the problem is further refined and decisions are made that limit the scope of the study. The objective hierarchy is further developed and a value system is created as a method for measuring each design alternative. Information is collected on satellite designs and satellite subsystems. Tradeoffs are performed to determined the best methods and components to be used in the alternatives. A model is created and design alternatives are generated. System analysis is performed on the alternatives using the value hierarchy, and results are generated. Sensitivity analysis is performed on the alternatives, and implementation recommendations are provided to the sponsor.

The third volume provides details on the tools developed to build a satellite and to analyze the design. There are three sections to this volume. The first section describes the model's philosophy and presents details on the purpose and operation of each module of the model. Mathematical formulae and module architecture are also described in this section. The second section is a user's guide to operating the model. Specific details of the sequence to be used and information required to run the model are provided in this discussion. The final section of this volume is the actual code of the model. The code is contained in an annex and is maintained by AFIT's Aeronautics Department at Wright-Patterson AFB, Ohio. The code can be provided to allow future modelers to understand and refine the work that has been accomplished.

Acknowledgments

The systems engineering design team would like to thank all the people who have provided their guidance, support, instruction, and personal time to ensure that the design study was a success. Special thanks go to the team's advisors, Lt Col Stuart Kramer, Maj Ed Pohl, and Dr Chris Hall. The team also realizes that it took more than just the team members to make the study meaningful and complete. Therefore, we wish to acknowledge the efforts of Lt Col Stan Correia, Maj Brad Prescott, Maj Scott Thomason, Doug Holker, Edward Salem, Lt Mike Rice, Capt Joel Hagan, Dave Everett, Col (ret) Edward Nicastri, Lt Col Brandy Johnson, Richard Warner, Linda A. Karanian, and the Space Warfare Center for the assistance and expertise they provided throughout our study. Finally, we would like to thank our families, who supported this effort with their patience and understanding: Sheri Carneal; Sedef and Sena Cokuysal; Rebecca, Austin and Travis From; Donna Krueger; and Coleen Robinson.

The Systems Engineering Team

Satellite Skunkworks

VOLUME III: TOOLS

PREFACE IV

ACKNOWLEDGMENTS VI

LIST OF FIGURES XI

LIST OF TABLES XIII

ABSTRACT XIV

1. MODSAT MODEL 1

1.1 INTRODUCTION AND BACKGROUND 1

1.2 MODSAT MODEL STRUCTURE AND FUNCTIONS 3

1.2.1 Basic Modsat Requirements and Development 3

1.2.2 Modsat High Level Structure 4

1.2.3 Data Structure and Flow between Modules 6

1.2.4 File Operations/Management 7

1.3 BUILD SATELLITE 8

1.3.1 Mission Module 8

1.3.2 Attitude Determination and Control (ADCS) 13

1.3.3 Electrical Power System (EPS) 17

1.3.4 TT&C and Data Handling *21*

1.3.5 Structures and Mechanisms *25*

1.3.6 Thermal *27*

1.3.7 Propulsion *28*

1.4 TEST SATELLITE 32

1.4.1 Scope *32*

1.4.2 Test Satellite Construction Algorithm *33*

1.4.3 Test Satellite Modules *33*

1.4.4 Test Satellite Limitations *39*

1.4.5 Test Satellite Future Work *39*

1.5 INITIALIZE OPERATING SYSTEM 40

1.5.1 Initialize Operating System *40*

1.5.2 System Reliability *44*

1.6 RUN TESTS AND SCENARIOS 47

1.6.1 Scope *47*

1.6.2 Run Tests and Scenarios Construction Algorithm *47*

1.6.3 Run Tests and Scenarios Modules *47*

1.6.4 Run Tests and Scenarios Limitations *50*

1.6.5 Run Tests and Scenarios Future Work *50*

1.7 ANALYZE DATA 51

1.7.1 Scope *51*

1.7.2 Analyze Data Construction Algorithm *51*

1.7.3 Analyze Data Modules *51*

1.7.4 Analyze Data Limitations *56*

1.7.5 Analyze Data Future Work *56*

2. USER'S GUIDE 58

2.1 SETUP INFORMATION 58

2.2 KNOWN PROBLEM 59

2.3 DISCLAIMER 60

2.4 MAIN MENU 60

2.5 PART I: DEFINE ORBITAL PARAMETERS 62

2.6 PART II: BUILD A SATELLITE 63

2.6.1 Payload *64*

2.6.2 Propulsion *64*

2.6.3 Structures and Mechanisms *65*

2.6.4 Attitude Determination and Control System (ADCS) *66*

2.6.5 Thermal Control *67*

2.6.6 Telemetry, Tracking, and Commanding (TTC) *68*

2.6.7 Electrical Power and Distribution *70*

2.7 PART III: TEST A SATELLITE 70

2.7.1 Total Spacecraft Mass *71*

2.7.2 Subsystem Component Overlap *72*

2.7.3 Power Constraints *72*

2.7.4 Display Satellite in 3D *73*

2.8 PART IV: INITIALIZE SATELLITE.. 74

2.8.1 Cost Estimating Relationships (CER) *75*

2.8.2 Reliability.. *76*

2.9 PART V: RUN SCENARIOS 77

2.10 PART VI: ANALYZE DATA.. 78

LIST OF FIGURES

FIGURE 1-1: LOGICAL FLOW OF MODSAT MODEL 5

FIGURE 1-2: MODSAT DATABASE STRUCTURE 7

FIGURE 1-3: OVERLAP CHECKING 35

FIGURE 1-4: LAUNCH VEHICLES AND CONFIGURATIONS 36

FIGURE 1-5: LAUNCH VEHICLE TESTS 37

FIGURE 1-6: AXIAL AND LATERAL LOADING 48

FIGURE 2-1: MODSAT MAIN MENU 61

FIGURE 2-2: SCREEN FOR ORBITAL PARAMETERS 62

FIGURE 2-3: SCREEN FOR BUILDING A SATELLITE 63

FIGURE 2-4: PAYLOAD BUILD MENU 64

FIGURE 2-5: PROPULSION BUILD MENU 65

FIGURE 2-6: STRUCTURES AND MECHANISMS BUILD MENU 66

FIGURE 2-7: ADCS BUILD MENU 67

FIGURE 2-8: THERMAL BUILD MENU 68

FIGURE 2-9: TTC BUILD MENU 69

FIGURE 2-10: ELECTRICAL BUILD MENU 70

FIGURE 2-11: TEST SATELLITE MENU 71

FIGURE 2-12: TOTAL MASS CHECK 72

FIGURE 2-13: POWER CONSTRAINT CHECK 73

FIGURE 2-14: MENU SELECTION FOR SATELLITE DISPLAY IN 3D 74

FIGURE 2-15: INITIALIZE SATELLITE SCREEN 75

FIGURE 2-16: CER SELECTION MENU 76

FIGURE 2-17: RELIABILITY CALCULATION SCREEN 77

FIGURE 2-18: RUN SCENARIOS MENU 78

FIGURE 2-19: DATA ANALYSIS SCREEN 79

LIST OF TABLES

TABLE 1-1: DATABASE MANAGEMENT FILES 8

AFIT/GSE/GSO/ENY/96D-1

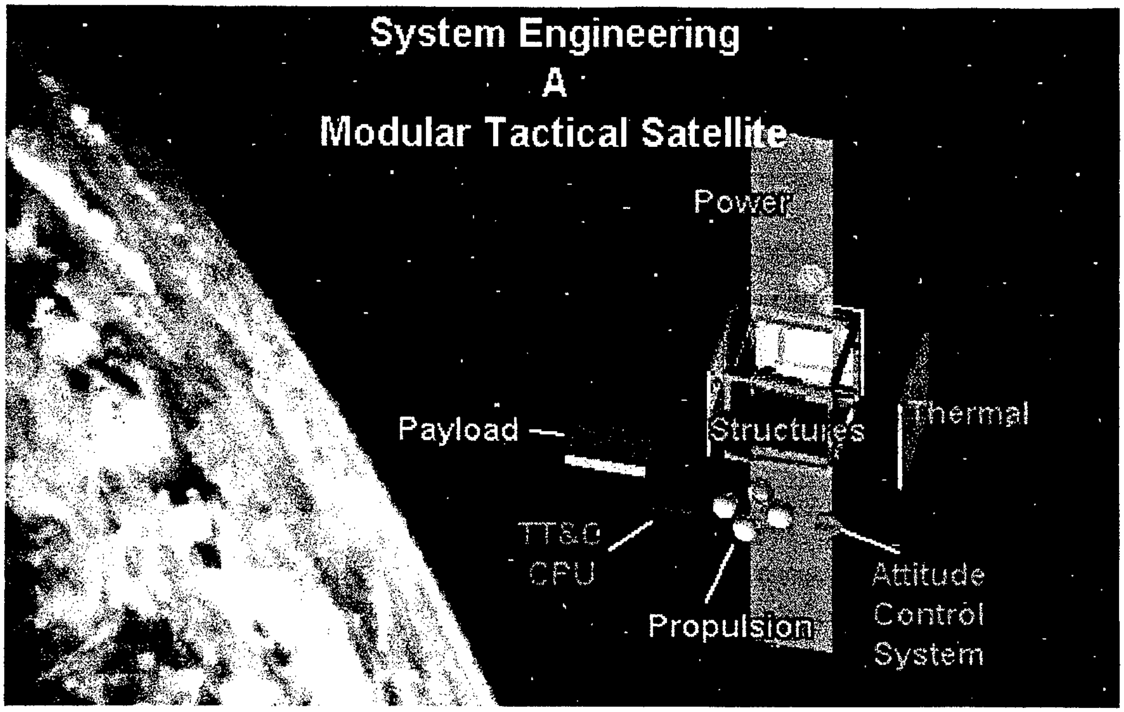
Abstract

A PRELIMINARY DESIGN OF A STANDARDIZED SPACECRAFT BUS FOR SMALL TACTICAL SATELLITES

Current satellite design philosophies concentrate on optimizing and tailoring a particular satellite bus to a specific payload or mission. Today's satellites take a long time to build, checkout, and launch. Space Operations planners, concerned with the unpredictable nature of the global demands placed upon space systems, desire responsive satellite systems that are multi-mission capable, easily and inexpensively produced, smoothly integrated, and rapidly launched. This emphasis shifts the design paradigm to one that focuses on access to space, enabling tactical deployment on demand and the capability to put current payload technology into orbit, versus several years by today's standards, by which time the technology is already obsolete. This design study applied systems engineering methods to create a satellite bus architecture that can accommodate a range of remote sensing mission modules. System-level and subsystem-level tradeoffs provided standard components and satellite structures, and an iterative design approach provided candidate designs constructed with those components. A cost and reliability trade study provided initial estimates for satellite performance. Modeling and analysis based upon the Sponsor's objectives converged the designs to an optimum solution. Optimum design characteristics include a single-string architecture, modular solar arrays, an internet-style command and data handling system, on-board propulsion, and a cage structure with a removable frame for easy access to subsystem components. Major products of this study include not only a preliminary satellite design to meet the sponsor's needs, but also a software modeling and analysis tool for satellite design, integration, and test. Finally, the report provides an initial implementation scheme and concept for operations for the tactical support of this satellite system.

Authors:	Capt Gerald F. Ashby
	Capt Darren J. Buck
	1st Lt Robert W. Carneal IV
	1st Lt Tansel Cokuysal
	1st Lt A. Tuna Donmez
	Capt James A. From
	Capt Todd C. Krueger
	1Lt Brian I. Robinson
Committee:	Lt Col Stuart Kramer, Maj Ed Pohl, Dr Chris Hall
Sponsor:	Lt Col James Rooney, PL/WS

1. Modsat Model



1.1 Introduction and Background

Models play an important role in solving complex and multiple variable problems; as such, it is a critical element of systems engineering. System Engineers are using models to perform various tasks. Some of these are listed below:

- Evaluate alternatives against a set of criteria, providing feedback about the alternative's performance.
- Generate sensitivity analysis information to take forward to the decision maker.
- Identify infeasible or low performing alternatives, so more time can be spent on those alternatives scoring better.

To use systems engineering in designing a standardized tactical satellite bus, the team was faced with many variables and the relationships among them. Satellite design by its very nature is quite complex, and trying to account for every detail in this preliminary study is infeasible. Modeling enabled the team to approach the satellite design at a high level, focusing only on the major elements of the spacecraft. The team also used the model to test and evaluate the performance of the alternative solutions.

Before considering modeling methods, it was important to revisit the problem definition, the objectives, and measures of effectiveness (MOEs). This ensured the modeling effort was fully relevant to solving the problem, and within the framework of the objectives the model should be able to evaluate the alternatives proposed to solved the problem.

Once the basis for the model was understood, the team outlined the scope and type of model necessary to meet the objectives. Starting from a high level, the team determined the model must be highly integrated and compatible; the best way to satisfy this requirement was to use one program for all of the modeling. Once the model produced a satellite design, it would be important to perform some level of environmental and integration testing on it. Those designs meeting the testing and integration modeling elements required data analysis to assist in the final evaluation of the alternatives ranked with each other.

Although these requirements for an integrated model appeared easy to fulfill, they were not. The research conducted on the Internet found satellite modeling software still geared toward specific subsystems. Discussions with aerospace companies confirmed our

findings; however, they mentioned how some companies are now using computer labs to bring subsystem experts together for satellite design sessions (Karanian and Warner, 1996). To satisfy all these modeling requirements, it was necessary to develop an in-house satellite design model, using Matlab, a mathematical and graphics software package.

To meet the "integratable, compatible, and adaptable" modeling requirement, Modsat (Modular Satellite) was constructed around a generic database structure format. By constructing all the subcomponents with the same generic database, compatibility amongst all subcompontents was achieved. Once the subcomponent databases were constructed, they were combined into a single satellite database, thus ensuring the overall satellite design was totally integratable. Modsat satisfies the "adaptability" requirement by allowing the satellite designer to modify Modsat in three ways:

- Correct, delete and/or expand the satellite database.
- Make changes to the Modsat code substructure to incorporate more detail analysis or changes in technology.
- Tie into other external programs

The Modsat model effort was quite extensive, addressing all the requirements discussed above. The remainder of the this document will cover the operation of Modsat.

1.2 Modsat Model Structure and Functions

1.2.1 Basic Modsat Requirements and Development

To begin the software development, the team further defined the modeling requirements for Modsat with the following qualifying factors:

- Design an integrated satellite with one modeling tool and construct it around a generic database.
- Use Matlab's built-in Graphical User Interface (GUI) to construct a "user-friendly" menu driven model.
- Perform initial testing of the integrated satellite design to ensure the design is feasible.
- Calculate satellite cost and reliability of each satellite design
- Perform launch and on-orbit testing to ensure the design will operate properly within the prescribed environment.
- Evaluate and score each of the satellite designs and provide an overall ranking against other designs.
- Perform database operations

1.2.2 Modsat High Level Structure

Using the modeling requirements listed above, Modsat development started with a high-level structure as shown in Figure 1-1.

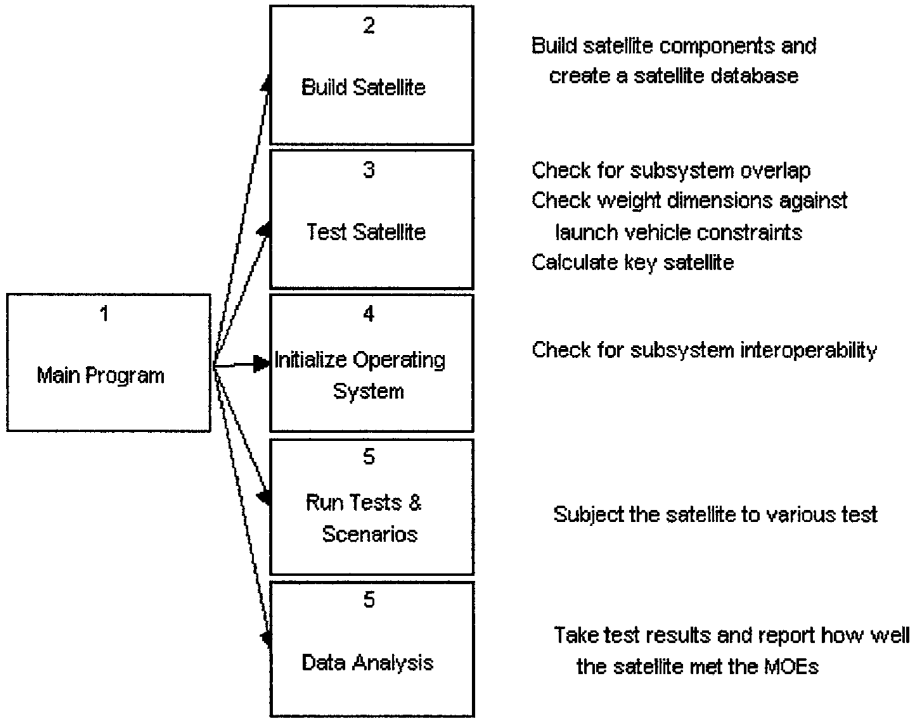


Figure 1-1: Logical Flow of Modsat Model

Normally, the Modsat model operates sequentially, starting from the top of the logic flow diagram and working down. Each step provides additional information about the satellite design, and at any given step the satellite design can be terminated for modifications, or the designer can start over. For added flexibility Modsat allows the user to run any of the applications at anytime, however, errors are likely, since some routines are dependent on previous calculations.

Build satellite: Build subcomponent databases and combine them into one integrated satellite database.

Test Satellite: Check the satellite database to ensure total satellite mass, center of mass, and sizing meet the launch vehicle constraints. If the satellite design fails any of these tests, the satellite design must be either abandoned or modified.

Initialize Operating System: Once the satellite design passes the "test" section, the satellite database is checked for subsystem interoperability such as power requirements. This section also calculates cost and reliability for the satellite design.

Run Tests & Scenarios: This section subjects the satellite bus to launch and orbit environmental testing to determine its overall performance.

Data Analysis: The satellite performance parameters are fed into data analysis either directly or indirectly and are then evaluated against a set of objectives. Each satellite design receives an overall utility score as well as an overall ranking with other designs. To complete the analysis and provide some variability in the results, sensitivity analysis can be performed by modifying and rescaling the top level objective weights.

1.2.3 Data Structure and Flow between Modules

Satellites like automobiles are a collection of subcomponents such as fuel tanks, power, frame, etc., and each subcomponent has material properties, mass, geometric shape, size, and placement within the satellite structure. Subcomponents may or may not require some level of power. To capture these attributes Modsat was constructed around a database structure shown below in Figure 1-2. After the satellite database has been created, Modsat can perform the necessary testing and evaluation.

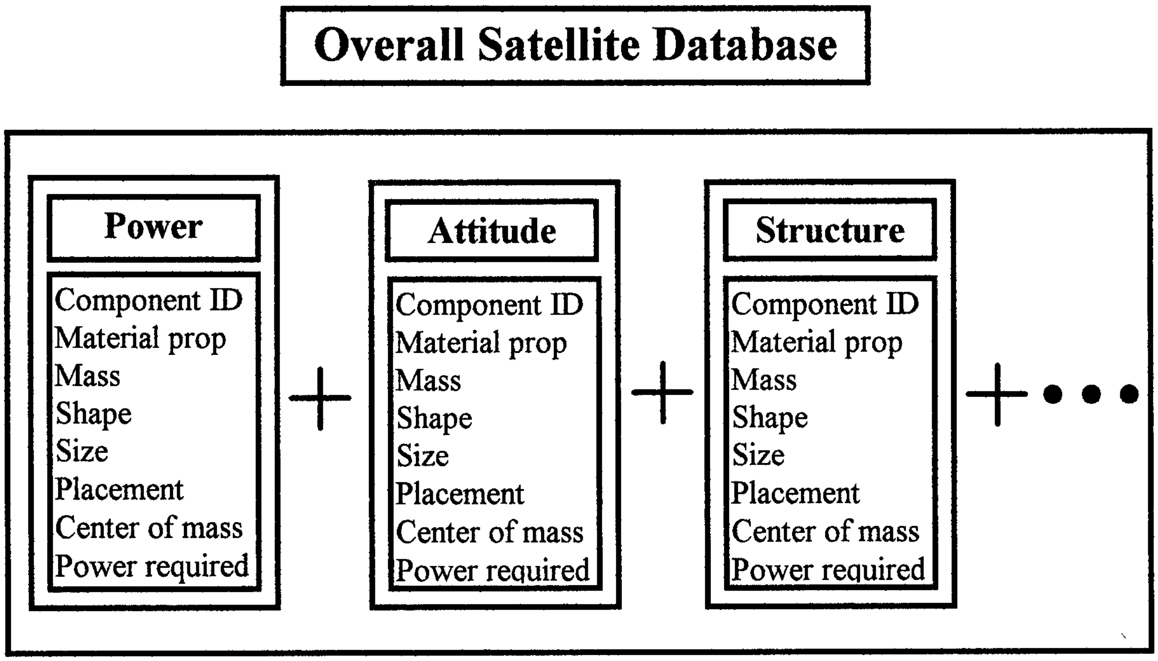


Figure 1-2: Modsat Database Structure

1.2.4 File Operations/Management

Database management within Modsat handles both saving and loading files for each subsystem. It also integrates each subsystem into an overall integrated database structure as discussed above. The quick save feature enables a user to batch save all subsystem files, including the integrated satellite file, by selecting one of eight pre-defined names for each area. This same concept is available in reverse, whereby the user may quick load files from disk. Subsystem files and the integrated satellite database can be initialized (erased) in order to start fresh.

Database management handles an array of 100 variables through centralized input/output operations. Because Modsat doesn't utilize the entire 100 variables database, it leaves room for future Modsat growth. These functions are performed by four files:

Table 1-1: Database Management Files

FILE	FUNCTION	INPUT VARIABLES	OUTPUT VARIABLES
annex.m	saves to existing file and appends	file_name array_name	
wrtover.m	saves and writes over existing file	file_name array_name	
wrtplus.m	saves and writes to data in file	file_name array_name	
readonly.m	loads file in read only mode	file_name	file_in

1.3 Build Satellite

All satellites are basically constructed with the same key subsystems such as power, communication, computer, attitude control etc., and all these subsystems are made up of subcomponents, which all have similar attributes such as size, shape, mass, etc. Modsat captured these attributes by creating subsystem databases with one generic database. This subsystem database contains all the subcomponents for that particular system. The following discussion provide clearer understanding of how each subsystem handled its particular database.

1.3.1 Mission Module

1.3.1.1 Scope

The defining component for any space mission is, of course, the payload (or mission module). The purpose of Modsat is to assist spacecraft bus designers; however, many characteristics of the mission modules desired to be flown on the bus will necessarily drive specific requirements for the bus's performance. For Modsat to properly evaluate

the performance of the bus against requirements, the characteristics of the mission module components must be modeled and/or input to Modsat. The mission module construction module of Modsat performs this function.

The original design request included treatment of electro-optical (EO), multispectral-imaging (MSI), LASER designation, and synthetic aperture radar (SAR) mission modules. Through the course of research, an EO with LASER (LIDAR) mission module, deemed a more generic approach to modeling a wide range of LASER applications, replaced the LASER designation mission. Also, due to the lack of availability of a large data set of specific, current physical specifications for individual mission module subcomponents, Modsat incorporated very generalized, first-order relationships for estimating the characteristics of mission module components. As each component is generated by the model, these physical characteristics may be edited by the user to tailor the mission module components to specific user needs. These features allow Modsat to evaluate bus designs against a virtual continuum of mission module requirements (as opposed to specific, pre-generated mission module packages of requirements).

1.3.1.2 Mission Module Construction Algorithm

The basic algorithm for the mission module incorporates modularized sections (each mission module type has its own section) comprised of user input screens, functions which generate initial mission module characteristics, and user edit/input screens for each component generated by the model. Each general mission module type has its own separate "pushbutton" and set of component-generating routines. Before any of these begin, and immediately following entrance into a mission module section, the initial

positioning of the centerline (in the axial orientation) for the mission module is calculated by extracting information from the structure database. Initial positioning of the mission module's centerline is otherwise determined by the user

As stated above, the estimators used to generate initial mission module component characteristics are first-order, initial estimators. The user has the opportunity to edit many of the mission module components' characteristics as they are generated in turn for each component; however, many are held fixed, such as positioning of the "sensor suite/base plate" component after initial position input at the start of each section. Different general mission module types are assigned specific component identification numbers within the database allocated to the payload components:

Electro-optical	100-119
Multispectral Imaging	120-139
LASER/LIDAR	140-159
SAR	160-179
Miscellaneous/nonspecific	180-199

1.3.1.3 Reinitialize Payload Database

For rapid repositioning of the mission module atop an integrated Modsat bus design, the main menu incorporates a payload database re-initialization button, which wipes out the previous mission module database (backs-up the old database). The mission module must then be reconstructed after this re-initialization. Since all components in the mission module are positioned off of the base plate, this process saves time over the tedious process of individually calculating and repositioning mission module components.

1.3.1.4 Individual Mission Module Sections

1.3.1.4.1 Electro-Optical (EO) Mission Module

A simple, initial simulation of the EO mission module first queries the user for the required diameter of the primary optics. Modsat then generates initial mass, power, size, and other component characteristics for three simplified mission module components: a generic "sensor suite" base, the primary optics and housing, and the secondary optics housing and tube. The user may then view the generated values and edit them if desired. The three basic components are assigned component identification numbers 100, 101, and 102, respectively.

1.3.1.4.2 Multispectral-Imaging Mission Module

MSI mission modules are similar to EO mission modules except the optics tend to be larger because the sensor suite supports six or more spectral bands with multiple detectors. With the exception of initially generated values for the mass, operating temperature, size, and power requirements, the MSI mission module construction algorithm and user interface flow is exactly that of the EO mission module mentioned above. Component numbers are 120, 121, and 122, respectively.

1.3.1.4.3 Electro-Optical with Laser (LIDAR) Mission Module

Modsat includes a mission module section for the LIDAR mission module, which is based on the EO primary optics diameter and the power of the LASER employed as the illuminator. In the event future Modsat users want to model a LASER designation mission, this section can serve as a generic LASER mission module. This will allow the user to create "exotic" LASER mission modules based on a specific LASER power

requirement. This section generates, in addition to the optical components similar to the EO mission module section, a LASER head and power supply/conditioner. Note that an EO with LASER mission module could also be used as a simple EO radiometer when the LASER is not being used for illumination of a target. Sensor and optical components are assigned component numbers 140, 141, and 142, and the LASER's component number is assigned 143.

1.3.1.4.4 Synthetic Aperture Radar (SAR) Mission Module

The SAR module requires the user to input the size of the array and the power requirement for the mission module. From these inputs, the module uses estimators for generating characteristics of both the array and the sensor suite. Again, this section is similar in basic algorithm to the other mission module sections. The two SAR components are assigned numbers 160 and 161.

1.3.1.4.5 Additional Tools: Mission Module Data Rate Calculator

A user-input driven data rate calculator provides as a "back of the envelope" or "first order" calculation of an EO mission module's data rate. This data rate calculator is taken from SMAD (Brodsky, 1992: 275) and provides a good, first-order estimation of the mission module's required data rate handling capacity.

1.3.1.5 Mission Module Limitations

The current version of the payload (mission module) components modeling section orients and associates the mission module components in specifically set orientations and positions, and this is determined by the initial placement of the base plate of the mission

module. Editing the individual components for sizing and positioning is somewhat tedious because it requires the user to perform hand calculations. Therefore, to reposition the mission module, the entire payload database must be reinitialized and rebuilt. This also requires the user to keep track of any characteristics edited out of the original estimations for those characteristics different from the original.

1.3.1.6 Mission Module Future Work

Future versions of the Mission Module Construction Module may include further development of the external interface to PCSOAP (Personal Computer Satellite Analysis Program by Aerospace). By doing so the user could obtain of revisit times, elevation angles (on a surface target), swath widths, and other useful information which may impact mission module requirements/characteristics.

More powerful editing capabilities, more detailed mission module modeling (more constituent components), and further determination of component characteristic estimators are the main refinements for future mission module versions. Future versions may also include more specific types of mission modules at various orientations other than fixed axial direction.

1.3.2 Attitude Determination and Control (ADCS)

1.3.2.1 Scope

The purpose of the ADCS code is to assist in designing a satellite's ADCS. It includes calculating the effects of external sources of disturbance torques, sizing the

actuators needed to compensate and provide for adequate spacecraft control, and choosing sensors and actuators based on the design specifications of currently available hardware.

1.3.2.2 Attitude Determination and Control (ADCS) Construction Algorithm

The ADCS model is designed to (1) compute disturbance torques due to the natural space environment, (2) compute the required torque capacity, angular momentum (H) storage capacity, and other performance parameters an actuator must meet to compensate for the disturbance torques and to control the satellite, and (3) allow the user to select ADCS hardware based on performance and description specifications of actual ADCS sensors and actuators stored in a database, that will meet performance requirements. The equations used in modeling the disturbance torques and in sizing the actuators come from Chapter 11.1 of Space Mission Analysis and Design (Wertz, 1992, pp. 340-366). Data on actual ADCS components comes from product brochures from numerous satellite contractors.

1.3.2.3 Attitude Determination and Control (ADCS) Modules

The ADCS model consists of 36 modules that together perform the functions described above. The model begins with a main menu that prompts the user to select one of four tasks: Calculate Disturbance Torques, Determine Actuator Size, Select Components, and Build ADCS. All except Calculate Disturbance Torques require some user interaction.

The module that estimates external disturbances calculates torque from four sources: gravity-gradient effects, disturbances from the satellite's interaction with the Earth's magnetic field, solar effects, and aerodynamics. It does so by reading critical satellite parameters (inertia matrix, solar array size, orbit parameters, etc.) previously calculated or from structures database. If these parameters are not present, default values are used. It then compares the four results and determines which is largest. That value is stored for use in sizing the actuators.

The next module is user interactive, and requires the user to input desired performance requirements. It then uses these, along with data calculated in the previous module, to determine the following: (1) the amount of torque an actuator must produce to reject the largest expected disturbance, (2) torque required to meet slew rate requirements, (3) the amount of angular momentum that must be stored per orbit by a reaction wheel or that must be generated by a momentum wheel in order to keep the satellite stable, (4) the force a thruster must supply in order to slew a reaction wheel or momentum wheel equipped spacecraft, (5) the amount of thrust per orbit needed to desaturate a reaction wheel.

The majority of the ADCS modules contain data on actual components. The user can select from five types of components: Earth sensors, star sensors, inertial measurement units, reaction wheels, and momentum wheels. The user then selects the type of component to review, which displays a brief description of each component. Upon selecting a particular component, more detailed information is made available. All data is

taken from the design and performance specifications of actual, currently available hardware.

The final option links the numeric database used by other parts of the overall Modsat model to draw and evaluate the proposed satellite. The user manually enters component data obtained from the previous section into the database, and the data is integrated into the attitude control system database.

1.3.2.4 Attitude Determination and Control (ADCS) Limitations

The ADCS model is limited in that the equations used only provide a preliminary estimate of the expected disturbances and the actuator sizing needed to handle them. Another inherent limitation is that the component database does not include the option of using sun sensors, magnetometers, and magnetic torquing devices, and in no way is it an all-encompassing list of the other types of components available. Further, advancements in technology will ultimately render those components listed obsolete. The ADCS module has no functions that would allow for preliminary design of the control laws. Finally, the module requires the user to write down component data and manually enter it into the database, whereas ideally this should be done automatically.

1.3.2.5 Attitude Determination and Control (ADCS) Future Work

Future efforts would primarily involve refining the disturbance and sizing calculations beyond the SMAD level, and inserting functions that allow the user to design control algorithms. Information on sun sensors, magnetometers, and magnetic torquing

devices would be included. Finally, component data would be automatically linked to the main Modsat database instead of having to enter it manually.

1.3.3 Electrical Power System (EPS)

1.3.3.1 Scope

The purpose of the EPS model is to size and construct the major EPS components. It incorporates the elements of the EPS design process discussed in the Tradeoffs chapter of the previous volume of this report. The major components include the solar arrays and batteries. The following components must also be created in the EPS model: power control unit (PCU, for distribution), regulator, and two solar array drive motors.

1.3.3.2 EPS Construction Algorithm

In the EPS model, the user is faced with four options which must be performed prior to entering items into the satellite database. The first option, "Define Power Drivers," allows the user to specify those system parameters that drive the design of the EPS. The second option, "Solar Arrays," leads the user through the process of selecting and rating the solar arrays. The third option, "Batteries," leads the user through the process of defining the energy storage requirements and selecting the batteries. The fourth option, "Mass Calculations," performs component mass calculations. Once those actions are performed, the user may select "Build EPS Components" to enter EPS component parameters into the database.

1.3.3.3 EPS Modules

1.3.3.3.1 Define Power Drivers

The model first prompts the user to specify the following system level parameters: satellite design life (years) and satellite peak power requirement (watts). It also displays the available area for the solar arrays (obtained from the structural model). Following this, it calculates the orbital period, TP, in minutes (Bates and others, 1971:33):

$$TP = \left(\frac{2\pi}{\sqrt{\mu}} / (60\,\text{sec/ min}) \right) a^{3/2}$$

where μ is the gravitational parameter, equal to 3.986012×10^5, and a is the semi-major axis, in km. The maximum eclipse period, TP_e, is determined by calculating the maximum percentage of orbit spent in eclipse. This is done by first determining the angular radius of the earth, ρ (deg), from the satellite frame of reference, at altitude (Wertz, 1992:103-104):

$$\rho = 70 - 10\frac{a - 200\,km}{800\,km}$$

where a is the orbital altitude and can range from 200 to 1000 km. The eclipse rotation angle, ϕ (deg), is estimated as $\phi = 2\rho$ (Wertz, 1992:104). Finally,

$$TP_e = TP(\phi/360)$$

The minimum daylight period, TP_d is calculated as TP - TP_e.

1.3.3.3.2 Solar Arrays

Given the available solar array area, the program prompts the user to select the type of arrays. The user may choose between silicon or gallium arsenide solar arrays.

After this choice has been made, the model assigns values for cell output power (W/m^2) and annual degradation.

Given the solar array area and cell power, the model determines the beginning of life (BOL) power production per unit area. Presently, a conservative value of 0.707 is used for inherent degradation (McDermott, 1992:397), while the worst-case sun incidence angle is set at 23.5 deg (McDermott, 1992:400). The lifetime degradation of the solar array is calculated, followed by end-of-life (EOL) power production per unit area. From this, the model calculates the EOL power output, and the mass of the arrays.

1.3.3.3.3 Batteries

The program prompts the user to specify whether average or peak loads will be used to size the batteries. For a conservative approach to the generic bus, since the load profile of a given mission module is unknown, the designer may want to size the batteries based on peak loads. If this is the case, the model will determine the required battery capacity to either handle eclipse loads, or to supplement the solar arrays in providing peak power, whichever is the more demanding requirement. The user must specify whether nickel cadmium (NiCd) or nickel hydrogen (NiH_2) batteries will be used.

The program continues by calculating the number of required charge/discharge cycles throughout the life of the satellite:

$$cycles = 525600(\min/year)\frac{SL}{TP}$$

where SL is the satellite life, in years. At this point, the user is prompted to enter the allowable depth-of-discharge (DOD), which is dependent upon the type of battery employed. DOD vs. cycle requirement curves for NiCd and NiH_2 batteries are found in

Space Mission Analysis and Design (McDermott, 1992:404). The program uses this value to calculate the total required battery capacity, both in watt-hours and amp-hours.

The user is then presented with a list of batteries to choose from, with characteristics such as energy density, capacity per battery, mass, and dimensions. An appropriate selection (type and number) must be made to provide the required capacity.

1.3.3.3.4 Mass Calculations

The program calculates and displays the following component masses (Reeves, 1992:319):

Mass of PCU = 0.02 kg * EOL power output

Mass of regulation equipment = 0.025 kg * EOL power output

1.3.3.3.5 Build EPS Components

The characteristics for each EPS component must be assigned numbers and entered into the subsystem database. Note that the solar array panels are entered in the structural model. A component representing electrical wiring must be created, and should reside within the central structural spine. This component should have a mass of between 1% and 4% of the dry bus weight (Reeves, 1992:319).

1.3.3.4 EPS Limitations

The EPS model only performs those calculations that were necessary to generate a high-level set of EPS components for Modsat alternatives. Solar array output is defined by the launch vehicle fairing dimensions and the geometry of the bus, as opposed to a stated average power requirement. Component choices are limited, and all component

characteristics must be manually entered into the database. The user must have a battery DOD vs. cycle requirement curve available.

1.3.3.5 EPS Future Work

A complete set of component choices (including full sets of characteristics) should be included for each major component. Upon selection of the appropriate components, the model should automatically enter the characteristics into the subsystem database.

1.3.4 TT&C and Data Handling

1.3.4.1 Scope

The purpose of the Telemetry, Tracking and Commanding (TT&C) module is to incorporate aspects associated with the satellite's communication system. This action includes entering values for characteristics of the components that make up the TT&C subsystem. These parameters are then used to help determine characteristic performance of the selected TT&C subsystem, specifically maximum supportable downlink rate.

1.3.4.2 TT&C and Data Handling Construction Algorithm

To build the TT&C database, Modsat permits the user to perform five different functions. These functions include the ability to: (1) estimate the size, weight, and power requirement of the components, (2) enter the type of modulation techniques the TT&C system will be using, including bit error rate and receiver temperature, (3) enter parameters for transmitter/receiver pairs, (4) enter information on sources of jamming to

the satellite, and (5) enter the parameters associated with a TT&C component. The majority of the data and formulae used to create the TT&C component section was derived from the Space Mission Analysis and Design (SMAD) text (Wertz, 1992:390). For those sections of Modsat devoted to communication, information was taken from the Principles of Communications Satellites by Gary Gordon and Walter Morgan, 1993.

1.3.4.3 TT&C and Data Handling Modules

The estimation for the TT&C's weight, size, and power estimations was developed to provide critical component parameters should the user not have access to specific information for TT&C components. Modsat allows users the option of designing the TT&C subsystem in parts or as an integrated system. Users are also provided the option of specifying the level of complexity desired in the subsystem design. Modsat allows the user to enter a known quantity of information about a subsystem design characteristic, i.e., weight, size or power. The other two unknown parameters are then calculated using an interpolation method based on the information from the SMAD section on Command and Data Handling. For instance, if the designer knows the weight of a transceiver, the power required and the size are displayed to the user without manual calculations.

The sections of Modsat pertaining to the communications type, transmitter/receiver information, and jammer information are additional features which allow the user to interface with The Aerospace Corporation's PCSOAP software package. In the communication section the user is allowed to enter information about the type of modulation technique employed (Phase Shift Keying, Binary Phase Shift Keying, etc). Given the specified bit error rate (BER), receive antenna's system temperature, and other

critical parameters from the transmitter/receiver PCSOAP calculates the optimum data downlink rate. This downlink rate is then used in the analysis section of Modsat.

The transmitter/receiver data is automatically loaded into the TT&C database as the user inputs various values for transmitter/receiver antenna diameters, transmission loses, atmospheric loses, type of receiver (airplane, ship, ground station), transmitter power, etc. Until the interface with Aerospace Corporation's PCSOAP program is established, these communication parameters within the TT&C subsystem database are not used.

The jammer module permits the user to enter similar information as the transmitter/receiver information section (antenna diameters, link loses, transmission power, etc). This data is used in PCSOAP to determine the satellite's communication system's ability to operate in a hostile environment. At the time of this report, this section of code was not required to determine the best satellite design.

Both transceiver/receiver and jammer information are entered as communication pairs. In other words for every transmitter there must be a receiver; otherwise, defining a communication system does not make sense for analysis purposes. This method of definition means the same antenna on the satellite may be redefined multiple times if multiple ground stations communicate with the satellite. This may be cumbersome, but provides the powerful feature of distinguishing variations in bandwidth and operating frequency from one ground station to another. PCSOAP uses these definitions to determine how well link budgets operate.

The last and most crucial aspect of building the TT&C database is specifying the component characteristics. Each component's attributes of size, shape, power

requirements, etc. has to be manually entered into the TT&C database. This database is comprised of all the subcomponents required for the construction of the TT&C subsystem. As with all other subsystem databases, the TT&C database will be merged into the overall Modsat database.

1.3.4.4 TT&C and Data Handling Limitations

When the user determines the size, weight or power of a component using the estimation function, the data has to be manually entered into the Construct TT&C component section of Modsat.

In order to make the PCSOAP interface work correctly, users are required to enter both the communications type and the transmitter/receiver information. If the information is not complete, PCSOAP will produce erroneous data and may not work.

1.3.4.5 TT&C and Data Handling Future Work

The Telemetry, Tracking and Commanding module performs the basic functions necessary to produce data for the analysis section. Including an automated database entry function would make the TT&C modules more user friendly would. The automated functionality could be in the form of a known TT&C listing where the user could highlight the components desired. These parameters would then be loaded in one step. This process could also be applied to the size, weight, and power estimation functions.

1.3.5 Structures and Mechanisms

1.3.5.1 Scope

The main constraint to designing a satellite bus structure is the launch vehicle's payload bay. Because the payload bay's dimensions and volume are fixed, the satellite's size and weight is considerably restricted. Therefore, the primary objective is to maximize the volume of the payload bay, while satisfying the weight constraint. The greatest difficulty in satisfying these competing objectives is creating a large but light satellite bus capable of meeting all structural loads placed on it during launch preparation, launch, and on-orbit operations.

1.3.5.2 Structures and Mechanisms Construction Algorithm

Modsat allows the designer to build a satellite bus with various building materials, number of sides, and overall satellite bus and solar panels construction parameters. Once the designer has entered these basics satellite bus parameters, Modsat automatically builds the satellite structure, the solar panels, and the woven insulation layer just below the top interface plate

1.3.5.3 Structures and Mechanisms Modules

To maximize volume Modsat creates the structure, solar panels, and mounting plates databases by querying the designer for the following inputs:

- Number of sides
- Construction material to be used throughout the satellite's construction

- Thickness and number or wraps to construct the solar wings
- Beam diameter and thickness
- Launch vehicle selection to determine solar wings maximum height in the payload fairing
- Number, thickness, and placement when constructing the mounting plates
- Structure reliability

Modsat structural design is iterative, allowing the user to continue changing the design until he/she is satisfied. Once the user is set on a particular structural design and has defined the mounting plate attributes, the structures database is automatically constructed as well as the solar panels and the thermal interface blanket, which are place in the power and thermal subsystem databases respectfully.

1.3.5.4 Structures and Mechanisms Limitations

Constructing a structural design has some limitations. For example, when using Modsat to create a structure, Modsat always creates solar panels in the "wrap-around" configuration, which discounts the use of folded arrays. Even though the designer can modify the number of wraps and solar wing thickness, the designer does not have the flexibility to modify the placement of the solar panels once the panels are built. Additionally, because Modsat creates a polygon structure made with cylindrical beams and circular mounting plates, other nontraditional structural designs are not possible at this time. Finally, the use of mounting plates precludes the use of any other mounting mechanisms.

1.3.5.5 Structures and Mechanism Future Work

To provide more designer flexibility, Modsat should allow the creation of non-standard cylinder-type designs. Because considerable engineering software is available to perform thermal and structural analysis on AutoCAD *.DXF files, Modsat such be modified to be compatible with AutoCAD.

1.3.6 Thermal

1.3.6.1 Scope

Thermal modeling is important to the satellite designer, ensuring the other subsystem's subcomponents do not exceed their temperature limits. Modsat allows the designer to manually construct either active or passive thermal systems. If later the satellite design appears to be thermally imbalanced, additional thermal devices can be added.

1.3.6.2 Thermal Construction Algorithm

During satellite thermal construction, Modsat allows the designer either incorporate insulation coverings around existing subcomponent components or to construction insulation panels to placed around the perimeter of the satellite design.

1.3.6.3 Thermal Modules

If the satellite designer knows a particular subcomponent is sensitive to thermal heating, the user can construct a insulation blanket to surround that component. Otherwise, the designer could create an active thermal heater.

1.3.6.4 Thermal Limitations

Although Modsat will allow the designer to construct a passive or active thermal system, it is not integrated with the thermal testing algorithm. The thermal module did not address the use of thermal coatings.

1.3.6.5 Thermal Future Work

Although passive thermal devices should be used to the greatest extend, some use of an active thermal system is likely, therefore, more research needs to be conducted to incorporate active systems within Modsat.

1.3.7 Propulsion

1.3.7.1 Scope

The main purpose of the propulsion model is to determine total volume, weight, cost, and the other parameters for the propulsion subsystem.

The propulsion subsystem model considers only the liquid propulsion options for the satellite bus. The model consider only tanks and engine which are the major parts of a propulsion system. The rest of the system such as pipes, transducers, valves, regulators, fittings and vanes, is out of scope of this study. However, they will be taken into

consideration as percentage (between 5-15%) of total tank volume and weight. Larson W.J. and Wertz, J.R (1995:660)

1.3.7.2 Propulsion Construction Algorithm

The algorithm for propulsion is very simple and uses basic equations for calculations which are explained in propulsion subsystem trade-off.

The program first calculates the propellant mass for the mission requirements. The mass of satellite, required ΔV, number of tanks, propellant type (the specific impulse of a propellant) are the drivers for mass calculation. The specific impulse of propellant is pulled from propellant database for the chosen propellant.

The propellant mass in each tank is calculated by;

$$m_p = m_f \cdot \left[e^{\left(\frac{\Delta V}{I_{sp} \cdot g}\right)} - 1 \right] = m_o \cdot \left[1 - e^{-\left(\frac{\Delta V}{I_{sp} \cdot g}\right)} \right]$$ Larson, W. J., Wertz, J.R. (1995:641)

where $m_f = m_o - m_p$: the final vehicle mass

m_o : initial vehicle mass

m_p : mass of propellant consumed

R : mass ratio (m_o / m_f)

I_{sp} : specific impulse

g : gravitational constant.

ΔV: required delta V during mission of satellite.

For mono-propellant systems the fuel mass will be equal to propellant mass. However, for bi-propellant systems the mixture ratio (by volume and by mass) of the chosen combination is pulled from database to calculate the fuel and oxidizer mass and volume.

For blow-down systems the pressurant gas volume and mass, and the blow-down ratio are used in calculations to get the total mass and volume. Finally, the empty tank weight is added to hold the propellant and pressurant. The weight of the empty tank is calculated by using the density of tank material, shape of the tank and the initial pressure of the tank. When the initial pressure is increased the weight of the tank increase to hold that amount pressure by increasing the thickness of the wall. During all calculations the titanium is taken to handle the hydrazine type propellants with a safety.

For regulated system weight and volume are calculated by same way except the pressurant gas weight and volume. Regulated systems keep their pressurization gas in separate containers. The regulated systems may have one or two pressurization tanks and more of the fuel or fuel and oxidizer tanks.

The tanks can be calculated as a combination of mono-propellant or bi-propellant and blow-down or regulated pressure fed system.

The user can chose the an engine and stick to the specific position on the bus. There are 10 off-the-shelf engine samples distributed between engine 0.45 N (0.1 lbf) to 444.8 N (100 lbf).

1.3.7.3 Propulsion Modules

The model for chemical propulsion gives options to set a basic structure. The playable elements of the propulsion model that drive the main modules, are;

- Number of propellant tanks.
- Type of propellant.
- Required ΔV and mass of the spacecraft.
- Type of pressurization system and pressure values.
- Shape of tanks.
- Type and size of the thrusters,

1.3.7.4 Propulsion Limitations

The propulsion code is written to design propulsion subsystem of a satellite bus design which will support small and tactical mission payloads. The early design trade-off dictated that the bus will be supported by a chemical propulsion with pressure fed system.

The main limitation is this model can not be used to design for other propulsion systems, such as electric propulsion, cold gas propulsion, or hybrid propulsion. Although it needs some modifications or rewritings to design these propulsion systems, the integration with the bus will not be problem. The modular style of architecture allows to design for all subsystem combinations.

The model is written to evaluate high level system architectures. So, the details for the propulsion system is not given by this model. The major and vital parts for a propulsion systems such as, tanks, pressurization systems, and engines are considered. During the calculations only major changes are considered and other details are kept the same for all alternatives such as, the pressurant gas is helium and all tank materials are titanium alloy for all alternatives. This is not a big limitation, the code can handle easily these kind of minor changes.

1.3.7.5 Propulsion Future Work

The pumpkin or doughnut shape (not exact shape) tanks should searched for optimal volume, weight, and pressure conditions.

Although this study is not a unique computer aided design generation for satellite, the effort and the leap in the model area is very big. For the future addition;

- The other propulsion systems should be coded.
- The propellant and engine database should be expanded.
- The code should be deal more details with respect purpose of the study.

1.4 Test Satellite

1.4.1 Scope

Once a satellite database containing all the subcomponents has been constructed, it will be necessary to ensure the overall satellite design meets specific weight and size requirements for the launch vehicle it will be launched on. This section covers those testing aspects of Modsat

1.4.2 Test Satellite Construction Algorithm

After the satellite subcomponents have been constructed and integrated into a satellite database, various modules calculate the key satellite's overall and subcomponent mass properties, CG values during launch and on-orbit, and inertial matrix during on-orbit operations. Next, Modsat uses these key parameters and the satellite database to perform the following subchecks to ensure:

a. Total mass does not exceed launch limits

b. The satellite's CG does not exceed launch vehicle tolerances

c. Subcomponents do not overlap each other

d. The launch vehicle payload bay dimension constraints are not exceeded

e. Maximum altitude for LEO is checked for given mass and inclination.

1.4.3 Test Satellite Modules

1.4.3.1 Total Mass Calculation

The following expression finds the total mass of Modsat design. To account for miscellaneous masses such as harnesses, wires, connectors, and propulsion piping 10% is added on top of component mass.

$$\text{Total\_mass} := \sum_{i} \text{component\_mass}_i + \text{misc\_mass}$$

1.4.3.2 Center of Mass Calculations for a Stowed and Deployed

These expressions below were used to determine the center of gravity during launch and on-orbit operations. From the satellite database each individual subcomponent's CG x, y, and z positions from an inertial fixed reference frame is multiplied by the subcomponent's mass and totaled. This is then divided by the overall mass to obtain the satellite overall CG position.

$$\text{X\_center\_of\_mass} := \frac{\sum_i \text{component\_mass}_i \cdot \text{x\_dist}_i}{\sum_i \text{component\_mass}_i} \qquad \text{Y\_center\_of\_mass} := \frac{\sum_i \text{component\_mass}_i \cdot \text{y\_dist}_i}{\sum_i \text{component\_mass}_i}$$

$$\text{Z\_center\_of\_mass} := \frac{\sum_i \text{component\_mass}_i \cdot \text{z\_dist}_i}{\sum_i \text{component\_mass}_i}$$

1.4.3.3 Inertial Matrix Calculations

Once the satellite's total mass and launch cg has been calculated, the following expression translates each subcomponent's inertial matrix and converts it to an overall satellite inertial matrix.

$$\text{I\_total} := \sum_i \left(\text{I\_original} + \text{component\_mass}_i \cdot \text{r\_x} \cdot \text{r\_x}\right)$$

I_original: represents the original inertial matrix for a given sphere, cylinder, cone, etc

component_mass: represents each subcomponent's mass found in the database

r_x: represents the transformation matrix converting the object's local frame of reference to the satellite's launch cg frame of reference.

1.4.3.4 Overlap Check

Although this aspect of the Modsat does not rely on the previous calculations, it requires the satellite database. After reading the satellite database, this module simplifies the algorithm for checking overlap by converting each of the geometric shapes to box dimensions shown below in Figure 1-3. Once all objects are converted, each corner of every box is checked to see if it resides within the dimensions of another box. After the overlap check has completed, this module will return the results along with displaying graphically which satellite components did or did not overlap.

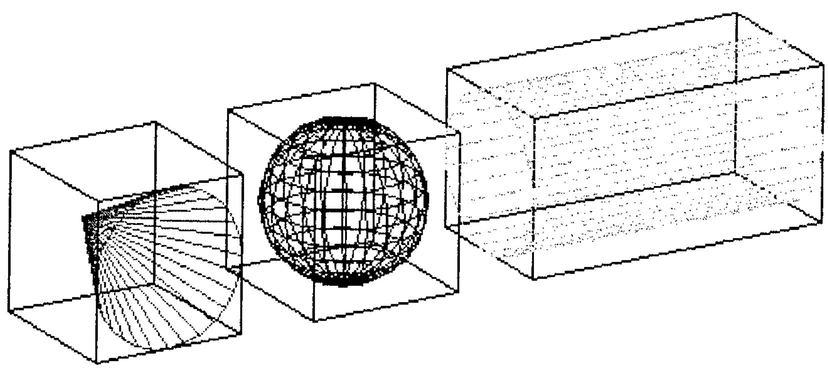


Figure 1-3: Overlap Checking

1.4.3.5 Launch Vehicle Check

This aspect of Modsat is ensures the constructed satellite will meet the launch constraints of the main launch vehicle, Pegasus-XL of Orbital Science Corporation. The following four checks are done; mass check, dimension check, center of gravity, and altitude check. Also this model incorporates the same launch vehicle checks for LMLV-1,

(Lockheed Martin Launch Vehicle) and Taurus (Orbital Science Corporation), for the configurations as shown in Figure 1-4. These additional two launch vehicle checks will not be used in the regular launch vehicle design procedure, since in this model each satellite configuration is designed for specifically Pegasus-XL launch vehicle. But if any launch vehicle check is not satisfied, then these tests may be used to get in advance information about compatibility to other two launch vehicles.

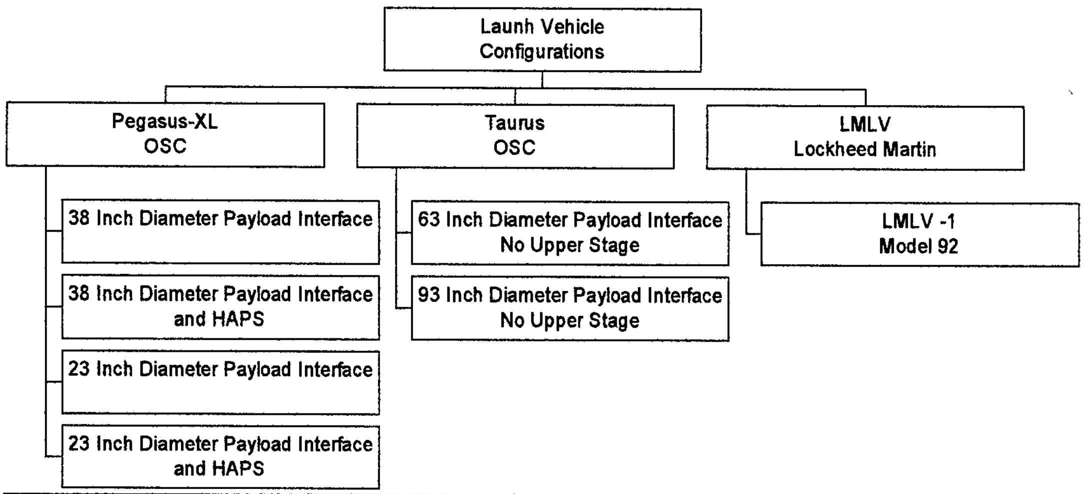


Figure 1-4: Launch Vehicles and Configurations

Basically, this model performs the four tests on the criteria mentioned above as shown in Figure 1-5. Each check is done sequentially. If the satellite does not satisfy anyone in the given order launch vehicle checks, the satellite preliminary design must be modified and after then the tests must be performed. Also, as subsystem components are being designed, the launch vehicle checks can be performed.

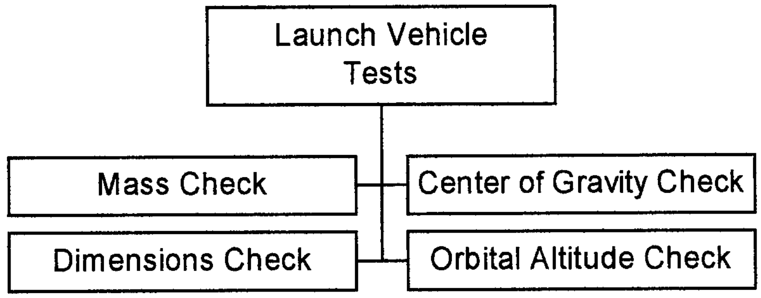


Figure 1-5: Launch Vehicle Tests

1.4.3.5.1 Mass Constraint Check

The total mass as calculated in section 1.1.5.3.1. is checked against the launch vehicle's maximum throw weight. For Pegasus-XL, since two standard adapters the following critical shears, 317.5 kg. (700 lbm.) for 23 inch adapter and 453.6 kg. (1000 lbm.) for 38 inch adapter, total mass is checked against the critical shear load of the selected adapter. For LMLV-1 and Taurus total mass is checked against the given maximum allowable masses in the user's guide.

1.4.3.5.2 Center of Gravity Constraint Check

The distance between the launch vehicle center line and center of gravity point for launch as calculated in section 1.1.5.3.2. is checked against the launch vehicle CG tolerances. This check is done in lateral and vertical axis for Pegasus-XL and just in vertical axis for LMLV-1 and Taurus

1.4.3.5.3 Launch Vehicle Dimension Constraint Test

In this test, satellite is checked whether it fits into the launch vehicle payload fairing dynamic envelope in three dimensions. It assumes the preliminary design is based upon the worst-case predicted ground, flight, and orbital loads. Also, for dynamic

interfaces, both manufacturing/design tolerances and payload dynamic deflections are accounted for.

The distance between each corner point of every component and center point is checked whether it is smaller than the radius of the dynamic fairing at that height for given launch vehicle configuration. If all corner points fit into the fairing envelope, than this design passes the launch vehicle dimension check for chosen configuration. All fairing dynamic envelope dimensions and specifications, including restrictions, are taken from the Launch Vehicle's Payload User's Guides. For each Pegasus-XL configuration, to check the fairing upper curved part, second order polynomial function is fitted to find the radius at given height.

1.4.3.5.4 Orbital Altitude Check

For Pegasus-XL second order multiple linear regression surface is fitted for altitude calculation. Predictor variables were inclination and mass, and response variable was altitude. Inclination value is taken from Keplerian orbit variables. Since the variation is not constant, error is approximately +/- 20 km. around 300 km. and +/- 60 km around 1,600 km. altitude. For LMLV-1 and Taurus, altitude is calculated for specific inclinations such as 28.5, 50, 90 degrees as taken from the performance charts from the User's Guides.

1.4.3.6 Display satellite in 3D

The ability to display the entire satellite design in 3-D is one of Modsat's most distinguishing features. After a satellite database has been integrated, the satellite designer

has the ability to selectively view all or some of satellite design in either the stowed or deployed configuration. Modsat also provides the designer more viewing tools. With Modsat the designer can rotate the satellite in any direction. To further viewing abilities Modsat was incorporated with a zoom in/out features. Because the 3-D model is color coded to reflect materials the satellite was built with, a materials color mapping was included as a cross referencing tool. To enhance the PCSOAP interface, Modsat can create PCSOAP *.vec models from the 3-D display.

1.4.4 Test Satellite Limitations

There are two limitations in the Modsat testing listed above. First, the major limitation the overlap checking algorithm is the "corner checks". This scheme overlooks any overlaps that may occur in the middle of the box. Second, although the 3-D display is very good, Matlab's shading objects algorithm does not always show the correct shading.

1.4.5 Test Satellite Future Work

To correct overlap deficiency, future work should concentrate on the breaking the satellite design into a 3-D grid. Although this configuration will require more computer processing, "nodal checking" will better ensure components do not overlap. To improve the 3-D display, future work could be done to utilized Matlab's ray-tracing algorithms to show true-to-life displays.

1.5 Initialize Operating System

1.5.1 Initialize Operating System

The purpose of the Initialize Satellite code is three-fold. The code ensures that a database exists for all subsystems, cost estimating relationships can be performed and satellite reliability can be calculated.

The Satellite Operating System (SOS) is the first function that the satellite design encounters when running the Initialize Satellite module. This code is analogous to having the central processing unit make a check to determine what subsystems are present within the satellite. The SOS performs a check to see if databases exist for all satellite subsystems. If all satellite component databases are not present, the test fails. A error message appears informing the user which satellite subsystem database(s) is/are missing. The user will be required to enter data for that subsystem and perform the Test a Satellite section before running the Initialize Satellite module again. If all checks are passed for SOS, the graphical user interface states that the satellite has been initialized and displays two buttons for the user to choose from. This section of the code is complete and is running.

The user will be presented with the Cost Estimating Relationship (CER) button and the Reliability button. These buttons and their functions are described below. Cost Estimation

1.5.1.1 Scope

Cost estimation plays a vital role in the evaluation of modern space systems for their overall utility to the Air Force; therefore, cost estimation efforts must not only be

current, but also must be robust. That is, the cost estimating relationships must address the various costing philosophies generally accepted by both the Air Force and industry. To emulate this broad range of philosophies, the Cost Estimation Module of Modsat will incorporate both Air Force and industry-generated estimation methodologies.

The purpose of this module of code is to determine the overall satellite costs in Fiscal Year 1996 Million Dollars (FY96$M). The module takes different aspects of a subsystem, usually mass, and forecasts the satellite cost using a cost estimating relationship (CER). The CERs are mathematical expressions generated from historical data of past satellite programs. The expressions used in this module come from the Aerospace Corporation's Small Satellite Cost Model (SSCM) Version 8.0 and the Space and Missile Systems Center Unmanned Space Vehicle Cost Model (USCM), Seventh Edition.

1.5.1.2 Individual Cost Estimation Modules

1.5.1.2.1 Aerospace SSCM

The SSCM divides input parameters into six CERs and then produces a weighted average of the individual calculations. The weighted average is based upon the inverse square of the percentage error associated with particular CERs.

The individual CERs for this model employ an additive-error method, and are derived from a linear least-squares relationship (assumption) of the parameter to the cost. The individual CERs will calculate cost, but are best used together (in the weighted average method mentioned above) to generate a more effective estimate. Version 8.0,

uses the General-Error Regression Model (GERM) to derive fewer CERs than the previous 17 CERs in version 7.4).

As an added feature for the Cost Module section, this module may be used as a stand-alone small satellite cost estimator. A user my input pre-generated values for the cost parameters -- if there is no initialized MODSAT database, or if the MODSAT database is incomplete, the initial values will appear either as zeros or bogus values. The user may then input the desired values for the input parameters and proceed with cost estimation.

1.5.1.2.2 SMC USCM

The Space and Missile System Center Financial Management and Comptroller office (SMC/FMC) provides the USCM to the Air Force. This cost model incorporates mostly military systems in its database, and does not specifically concentrate on smaller satellites.

This cost model provides four complete models, any of which are applicable to a given project, given the preference of the designer and the detail desired. The first two models are based on subsystem-level detail (provides 30 CERs), and the last two (much more detailed -- 70 CERs) are designed for component-level detail. At both the subsystem and the component levels the CERs are based on either the Minimum Percentage Error (MPE) or the Minimum Unbiased Percentage Error (MUPE) regression techniques. CERs from these different techniques cannot be used interchangeably; however, whole model estimates can be compared, providing a more robust (multi-methodological) approach.

The MPE technique seeks to improve on the ordinary least squares (OLS) approach by incorporating error as a percentage of estimation across the predicted range -- reflecting "reality" more closely than does a uniform dollar amount across the same range. The MUPE technique seeks to eliminate (by iteration) biases generated by a GERM-simultaneous optimization technique. Depending on the preference of the user, either set of CERs may be used to estimate costs -- and may be compared.

The MPE cost model approach, however, was judged to be the more suitable of the two methodologies for inclusion in the initial version of MODSAT due to the fact that 1) the MPE method produces the lowest percentage error, and 2) the error associated with MPE will always be positive in bias; therefore the cost will always be overestimated if in error. For the scope of the preliminary design study, this method is more appropriate, given time constraints on coding and debugging. Future versions of MODSAT may include the MUPE CERs for quick comparison of cost estimates.

Like the SSCM section, the USCM MPE cost estimation section has a stand-alone capability. whether or not the MODSAT database has been fully, partially, or not initialized at all, the user may input desired values for the individual cost parameters.

1.5.1.3 Cost Estimation Limitations

To use the CER module without having first integrated a Modsat database, the user, after initially calling the Modsat procedure, must type "bld_cer" in the MATLAB command window.

The CER module does not (as of 28 October 1996) include the MUPE section of the USCM subsystem-level CERs.

Care must be taken in the interpretation of the results of the different cost analysis models. While the USCM has different costs associated with recurring and nonrecurring costs, the SSCM rolls both of these costs into an aggregate figure to arrive at a single overall cost per satellite bus. The two cost models are not interchangeable. No cost estimation is included for use in mission module cost estimation.

1.5.1.4 Cost Estimation Future

Future features of the CER module will include the MUPE CERs of the USCM, and, as the component-level modeling capabilities of future versions of the Modsat design tool become more detailed and powerful, the CER module may also incorporate the MPE and MUPE component-level CERs, provide a more accurate estimate of costs.

1.5.2 System Reliability

1.5.2.1 Scope

The purpose of the Reliability modules is to compute the reliability of the Modsat design. It does this by prompting the user for mean time between failures (MTBF) for the seven major subsystems (TTC, ADCS, EPDS, Propulsion, Structures, Thermal, and CDH), mission length, and redundancy level of each subsystem. It then calculates subsystem reliability and combines these values to find the overall system reliability.

1.5.2.2 System Reliability Construction Algorithm

The Reliability model supplies MTBF values, in hours, based on historical satellite failure data, but at the same time allows the user to change any and all of these values to suit his or her particular circumstances. It also supplies default values for mission length in months (12 months) and redundancy level (single string, allowing up to triple-string), these are also tailorable to a user's needs. It then uses "stand-by" systems model assuming perfect switching between identical units, represented by a Poisson distribution, to compute the success probability (i.e. reliability) of a subsystem. In the case of a single unit, the equation to compute subsystem reliability is (Ramakumar, 1993:196)

$$\text{subsystem reliability} = e^{-t\,/\,MTBF}$$

where t is the mission length in hours (the algorithm automatically does the conversion from months to hours). In the case of one main and one backup, the equation is:

$$\text{subsystem reliability} = e^{-t/MTBF}(1 + t\,/\,MTBF)$$

In the case of one main and two backups, the equation is

$$\text{subsystem reliability} = e^{-t/MTBF}(1 + t\,/\,MTBF + (t\,/\,MTBF)^2\,/\,2!)$$

The algorithm could have been extended to include any number of backups, but the possibility of more than two backups is not likely.

After computing the reliability of each subsystem, the system reliability is computed by combining, in series, the subsystem reliability values. This is done because all subsystems are critical to satellite operation, and a failure by one would result in mission failure. The equation to compute system reliability is:

system reliability = $R_{TTC}(t)*R_{ADCS}(t)*R_{EPDS}(t)*R_{CDH}(t)*R_{Propulsion}(t)*R_{Thermal}(t)*R_{Structural}(t)$,

where R(t) is the reliability of the individual subsystems.

1.5.2.3 System Reliability Modules

There are only two Reliability modules. One creates the graphical user interface required to interact with the Reliability model, the other actually does the computations.

1.5.2.4 System Reliability Limitations

The major limitations in the Reliability algorithm lies in it's assumptions, and in it's scope. The assumptions used in creating the algorithm are (1) perfect switching between redundant units, and (2) standby redundancy. The algorithm makes no provision for changing either of these assumptions. The scope of the model is limited in that it only allows the user to go down to the subsystem level, instead of the component level, where redundancy is typically employed.

1.5.2.5 System Reliability Future Work

Future upgrades to the Reliability algorithm deal primarily with the two assumptions previously detailed. Specifically, the user would be asked whether or not perfect switching reliability should be assumed, and prompted for a switching reliability value if the answer is no. The model would be further developed by giving the user the option of breaking each subsystem down to the component level, and inputting values for MTBF and redundancy at that level.

1.6 Run Tests and Scenarios

1.6.1 Scope

Run tests and scenarios refers to all the tests necessary to ensure the satellite will endure launch and on-orbit environments and still operate to its prescribed operating requirements.

1.6.2 Run Tests and Scenarios Construction Algorithm

To ensure the satellite will operate in the space environment, launch vehicle, structural, thermal, and communication and data handling scenarios for the launch and on-orbit environment were developed. Although these testing algorithms are not all inclusive, all utilize the satellite database to extract key information for the particular test. Additionally, running testing scenarios may require orbit, mass, and cg parameters already calculated.

1.6.3 Run Tests and Scenarios Modules

1.6.3.1 Structural Test

Although using finite elements provide the best means of checking structural integrity of design, Modsat performs only "back of the envelope" calculations for structural strength. Therefore, the structural integrity is checked by applying an axial load to the satellite bus to determine how well the structure will perform under a 13 g axial load and 9.85 lateral g launch environment illustrated below in Figure 1-6. Also, Modsat checks the satellite bus's natural frequency and amount of deformation due to axial and lateral loads.

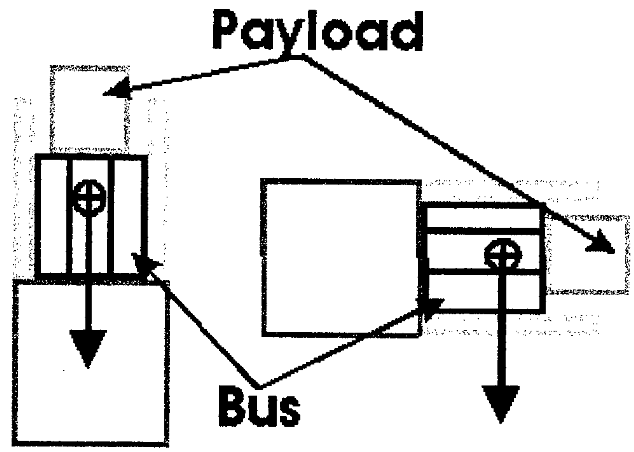


Figure 1-6: Axial and Lateral Loading

During structural testing of a satellite design the designer has two options. He can use the Modsat database to use the mass and cg parameters of that design or come up with his own tailored one. The designer also has the additional option to select the "g" loading and factor of safety to be tested against. Once these parameters have been loaded, Modsat performs the structural analysis and reports how well that structure performed. To enhance the output, Modsat uses color to flag whether or not a design passed a particular aspect of the testing.

1.6.3.2 Thermal Test

Modeling thermal conditions in any system design poses as one of the major difficulties. Because satellite thermal analysis must include conduction, radiation, and reflection between the satellite components, the sun, and the earth, finite element more closely models the temperature variations throughout the satellite. To simplify the thermal analysis only internal power, direct sunlight, and earth-emitted radiation will be modeled with the following equation:

$$Q\_interal + Q\_earth\_emitted + Q\_sun := \alpha \cdot \varepsilon \cdot Surface\_area \cdot Component\_temp^{4}$$

This module displays the temperature of each satellite component graphically at changing sun and earth angles throughout the orbit. This module assumes each satellite component is exposed to full sun and earth radiation and is protected only by some combination of thermal coatings, blankets, or heaters. Although this analysis does not fully represent true thermal conditions, it provides conservative estimates.

The thermal testing code is working, but it does not yet reflect changing sun angles and earth angles throughout the orbit.

1.6.3.3 Launch PCSOAP

The satellite model created in Matlab writes to "modsat.orb" in Matlab's working directory. Included in this file is a reference to "modsat.vec", a physical structure of the satellite created in "Display 3-D Satellite", so the satellite can be visually depicted in the simulation. PCSOAP, written by Aerospace Corporation, is used to simulate a given satellite in its designated orbit. Analysis functions can be performed, to include jammer and radio frequency propagation, coordinate frames, stabilization, sensor field of views, etc. Information gained in this simulation is saved by the user as "report.txt" which is then used by Matlab. This data is used to determine maximum data downlink rate.

At this time no code is written to provide sensor fields of view corresponding to placement of sensor packages on Modsat, such as star trackers and payload field of view. This feature is possible within PCSOAP and would be useful to help determine functional performance of the satellite design. Furthermore, code is not provided to cause solar panels to track with the sun, but is another feature available within PCSOAP with correct

stabilization and coordinate system definitions tied to the proper objects. PCSOAP provides a useful simulation tool, but its full extent is not utilized and it may be worthwhile expanding code in the future to extract maximum analysis from this software package.

1.6.3.4 Launch Vehicle Test

Once the designer has selected an orbit and has performed the mass calculations, Modsat will provide useful graphical information about the amount mass remaining for the payload, depending on the orbit launched into and the Pegasus launch vehicle selected.

1.6.4 Run Tests and Scenarios Limitations

Programming a finite element is outside the scope of this project.

1.6.5 Run Tests and Scenarios Future Work

As mentioned before the ability to create a satellite design and create a *.DXF AutoCAD file for export would be a tremendous improved to Modsat. Doing so would allow a designer to take advantage of the many software tools geared to using AutoCAD files.

Although the testing the satellite bus by the methods mentioned above will provide some indication of structural strength and thermal conditions, the structural an thermal analysis through AutoCAD conditions should be investigated.

1.7 Analyze Data

1.7.1 Scope

The purpose of this section of code is to evaluate how well a satellite bus design meets the requirements set forth in the Value System Design. By running a specific satellite design alternative through the objective hierarchy, a quantitative score can be assessed to that alternative. This score will permit a rank ordering of the alternatives that can be presented to the decision maker.

More importantly, this section of code permits the design team to perform sensitivity analysis on alternatives. By changing the weighting values assigned to the different objectives and subobjectives, the decision maker can assess which design alternative is more robust in different situations.

1.7.2 Analyze Data Construction Algorithm

The Data Analysis section has been developed to perform many different functions. Users are able to input pairwise comparison survey results to calculate objective weights, specify desired utility functions and risk preferences for the measures of effectiveness (MOEs), modify the utility functions, and enter/overwrite MOE data values. The code also allows the user to save the data analysis and perform sensitivity analysis with the saved data.

1.7.3 Analyze Data Modules

Objective weights are calculated using the data from the pairwise comparison surveys. The code permits the survey results to be directly entered into a database within

Modsat. Users can either add data to an existing database or create a new database. Once the data has been entered, the software calculates the objective weights in the following fashion. First, a geometric mean is applied to the comparison scores for each objective. A comparison score is the value that indicates how well an objective rated when compared to another objective. For a particular objective, the geometric mean requires each comparison score be multiplied by one another. The number of comparison scores is dictated by the number (n) of objectives being compared to one another. The result is a comparison score product. The nth root of this product is taken. This process provides the code with a resultant score (geometric mean) for the objective.

The resultant scores of the objectives being compared to one another are calculated. The sum of these resultant scores is used in the calculations to determine objective weights.

To determine the objective's weight, the objective's resultant score is divided by the sum of the resultant scores. This action results in a normalized value and will be equal to or less than one (1). By adding the normalized scores for each objective, the sum will add to one.

As more surveys are added to the database, an objective's weight is determined by taking the average of the weight in Modsat's database and the new weight calculated from the survey.

To perform analysis on a design alternative, data must be entered into the different measures of effectiveness. The Modsat code provides raw data for some measures of effectiveness and requires manual input for others. Measure of effectiveness data can be

accepted either in raw data form or as a scaled value. The Data Analysis code also provides the option of using the raw data produced by the Modsat code or allows raw data to be entered manually.

The Data Analysis code is configured to only accept values between 0 and 5. When it receives a value in this range, it converts the value to a utility value. This forces the code to convert raw data points such as Vehicle Mass into a scaled value. The Data Analysis code requires that a database be present with at least two satellite designs before it can convert raw data points into a scaled values. This occurs because there is no method of determining whether a raw data point should receive a good score (5) or a bad (0) score without having a reference. The code performs this function by assigning the highest value a 5 and the lowest value a 0 for cases where a high raw data score is considered 'better'. For cases where a low raw data value is best, such as cost, that data point receives a 5 and the highest raw data value receives a 0. If the database has three or more designs present, a proportional scoring routine is called to scale the raw data for values between 0 and 5

Another feature in the code is the user's ability to manually enter a raw data value if the user determines that the current value is incorrect or needs to be modified. Likewise, if the user prefers to use a scaled value instead of the raw data value, this option is also available.

As mentioned before, some measures of effectiveness require manual inputs. These data are generally items that cannot be calculated or produced mathematically. Additionally, data may not be physically available to be entered as a raw data point. In

these instances values will be entered into a MOE as a scaled value. How does one determine the time it will take to bring a satellite design to full rate production if the satellite is still in the design stage? The answer to this question requires the subjective judgment of the user. The answer may not be days, weeks or even months, but the user will have an idea of how a particular design will perform given the user's background and experience. To make it easier on the user, definitions are provided to the assist the user in determining the proper scaled value to assign to a design alternative.

During the actual running of the analysis code, each measure of effectiveness converts the scaled value between zero and five (0 and 5) into a utility value between zero and one (0 and 1). According to the Modsat code, a scaled value of 5 is always considered the best value, while a 0 is considered the worst. Likewise, a utility value of 1 offers the user the most utility and a 0 utility value offers no utility to the user. (Value System Design provides more details on utility functions.) The utility value is multiplied by the measure of effectiveness' objective weight and is propagated up the hierarchy.

The Data Analysis code permits the user to specify a desired utility function and risk preference for each measure of effectiveness. There are three predefined utility functions within the code. They model whether a user is risk neutral, risk averse or risk seeking (Clemen, R., 1996:459-502)

The risk neutral preference simply provides a constant proportionality between the utility value of 0 to 1. As depicted in the table below, a scaled value of 5 is converted into a utility value of 1, a scaled value of 0 is converted into a utility value of 0, a scaled value of 2.5 is converted into a utility value of 0.5, etc.

Scaled value	0	1	2	3	4	5
Utility Value	0.00	0.20	0.40	0.60	0.80	1.00

To model a risk averse attitude, the code's preset function do the following conversions:

Scaled value	0	1	2	3	4	5
Utility Value	0.00	0.30	0.55	0.75	0.90	1.00

A built in function of the code allows the user to modify the utility value if the user does not feel that this function models risk correctly. The utility values can be modified easily.

Likewise, the risk seeking profile follows a similar conversion scheme and be changed by the user to reflect the proper risk seeking attitude.

Scaled value	0	1	2	3	4	5
Utility Value	0.00	0.05	0.15	0.30	0.55	1.00

Once the data analysis has been performed, the top level objectives and the design alternative scores are presented to the user for all alternatives included in the database. The code also provides functions for saving the data used in the analysis and the actual data analysis results. By saving this data, the user is able to perform sensitivity analysis on the design alternatives. Sensitivity analysis is performed by modifying the weights associated with the top level objectives and re-running the analysis. The code is set up to allow the user to modify one top level objective weight at a time. When one weight is modified, the remaining weights are recalculated to ensure the weight sum is equal to 1.

During the recalculation, proportionality between the remaining weights is maintained. The code also permits the user to save the sensitivity analysis.

1.7.4 Analyze Data Limitations

When examining different design alternatives, it is important to check the raw data values for the MOEs. If a MOE consistently has an input that is the same for each design alternative, the raw data values will not be converted to a scaled value. During the conversion process, a 5 is trying to be associated with the best value and a 0 is trying to be associated with the worst value. A situation arises where the code tries to divide by 0. This will cause the code to produce erroneous data. The workaround to this problem is to manually input a scaled value in place of the raw data. The scaled value must be the same for each alternative. This prevents the divide by 0 problem.

1.7.5 Analyze Data Future Work

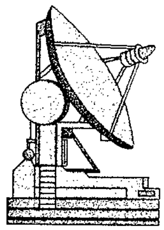
The Value System Design is fixed within the code. If a user wishes to alter the hierarchy and create a new objective tree, that function is not possible without rewriting the code. Future work could enable a user to dynamically select a value system design architecture, then analyze a set of satellite design options against it.

Sensitivity analysis is limited to altering weights of top level objectives only. Future work should allow the user to perform a broad range of sensitivity analysis tests.

Graphical analysis results are not provided, nor is automatic rank ordering such that the best alternative is listed first, the second best, and so on. Results are displayed by satellite with their scores in each of the five top level objectives along with a total score.

Such a table of numbers is not necessarily revealing, and thus graphical results would provide a better feel for the analysis.

2. User's Guide



WELCOME TO MODSAT VERSION 1.0

2.1 Setup Information

It is best to use at least a 486x or better running on a minimum of 8 Mb of RAM.

ModSat code was written using Matlab version 4.2c.1, made by Mathsoft ™. Ensure your copy is at least this version or better, which can be checked by typing "ver" at the Matlab command prompt and pressing <Enter>.

In order for Matlab to properly access all ModSat files, the following paths must be added to "matlabpath" in your MATLABRC.M file:

```
';c:\matlab\workarea\adcs',...
';c:\matlab\workarea\analysis',...
';c:\matlab\workarea\eps',...
';c:\matlab\workarea\integrte',...
';c:\matlab\workarea\payload',...
';c:\matlab\workarea\propulsn',...
';c:\matlab\workarea\scenario',...
';c:\matlab\workarea\structrs',...
';c:\matlab\workarea\thermal',...
';c:\matlab\workarea\ttc',...
';c:\matlab\workarea\file_ops',...
';c:\matlab\workarea\cer',...
```

```
';c:\matlab\workarea\lnchveh',...
';c:\matlab\workarea\draw_3d',...
';c:\matlab\workarea\reliblty',...
';c:\matlab\workarea\test',...
';c:\matlab\workarea\objects',...
';c:\matlab\workarea\overall',...
```

The above list may need to be modified to reflect your local drive and directory structure. Without this change, the program *will* crash.

ModSat interfaces with PCSoap ™ version 8.x written by Aerospace Corporation, so be sure a legitimate copy is installed on your computer before running ModSat. Also, make sure PCSoap is in your path command in your CONFIG.SYS file; otherwise, any interfacing between the two programs will be unsuccessful.

To start the program, at Matlab's command prompt type "modsat" and depress <Enter>. If all configurations are correct, you should get the main menu screen.

2.2 Known Problem

ModSat code does not run very well on any machine using Windows 95. More than one computer was tested with the same code, and independent machines exhibited the same problems: sometimes sections of code would run, and other times it would not. Windows 95 seems to have a pathing problem when running Matlab. Users should run Modsat code on a machine using Windows 3.x. Mac machines also work well with the code.

2.3 Disclaimer

The user is assumed to know what is required in the process of creating a first cut design of a satellite. This program does not give tutorials on satellite construction, and is intended as a design tool for analyzing the potential value of two or more satellite designs. At least two possible satellite designs are required since raw scores, such as cost, must be racked and stacked. A raw score by itself has no real value since there is no useful measure for "good" or "bad."

Beginning design of a useful satellite concept can be difficult, and the ModSat code is intended to assist with creating designs. Once a working satellite is defined, the initial design can be refined by other tools of the user's choice.

2.4 Main Menu

The opening screen shown in Figure 2-1 contains six areas which need to be completed in the order listed.

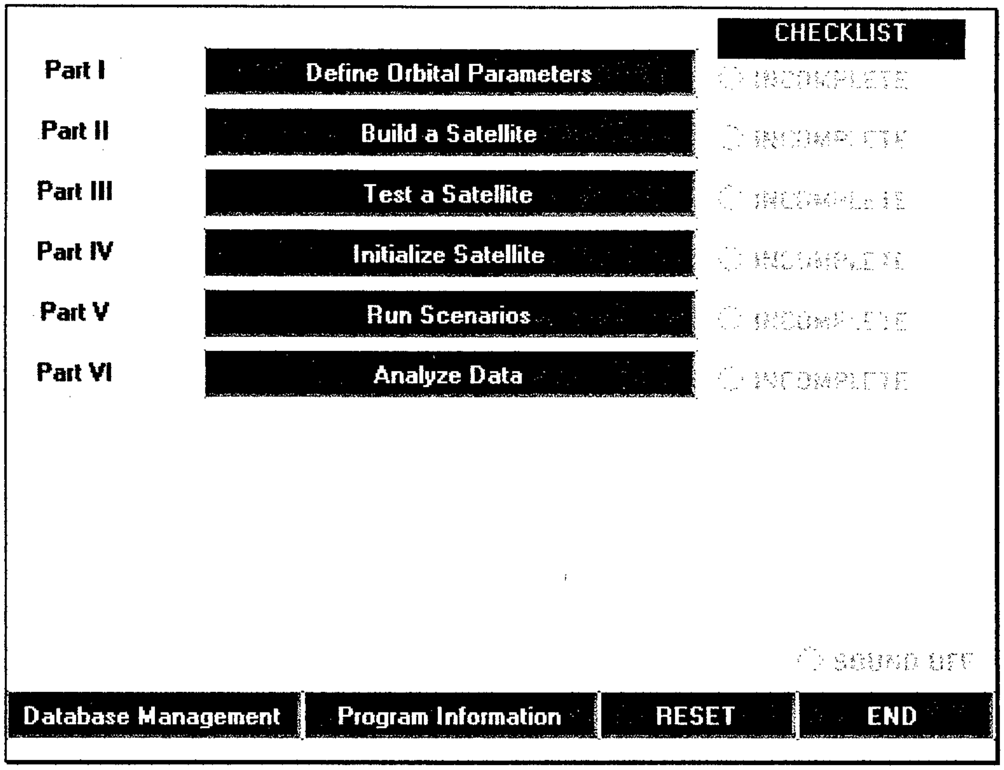


Figure 2-1: ModSat Main Menu

The six areas provide the following:

1. Define Orbital Parameters: enter ModSat's Keplerian orbital elements
2. Build a Satellite: construct subsystem areas, along with the capability to edit components
3. Test a Satellite: perform mass check, overlap check, power constraint check, as well as display ModSat in 3D
4. Initialize Satellite: simulated power on self-check, with links provided to cost estimation relationships and reliability if all necessary subsystems are on line; component cost list provided
5. Run Scenarios: provides structural check, thermal check, PCSoap interface, and launch parameters

6. Analyze Data: enter scores either as raw data or slider bar input on measures of effectiveness in value system design; provide comparative analysis when two or more designs exist

An interactive checklist to the right of the push buttons enables the user to toggle between a status of "incomplete" or "done." "Database Management" provides file loading, saving, and initialization features. "Program Information" lists relevant ModSat data. "Reset" performs reset of ModSat without quitting Matlab. "End" quits ModSat and closes out Matlab. For computers with sound cards, "sound off" can be toggled to "sound on," but will generate an error if a sound card is not detected.

2.5 Part I: Define Orbital Parameters

Figure 2-2 shows the screen for entering ModSat's orbital elements.

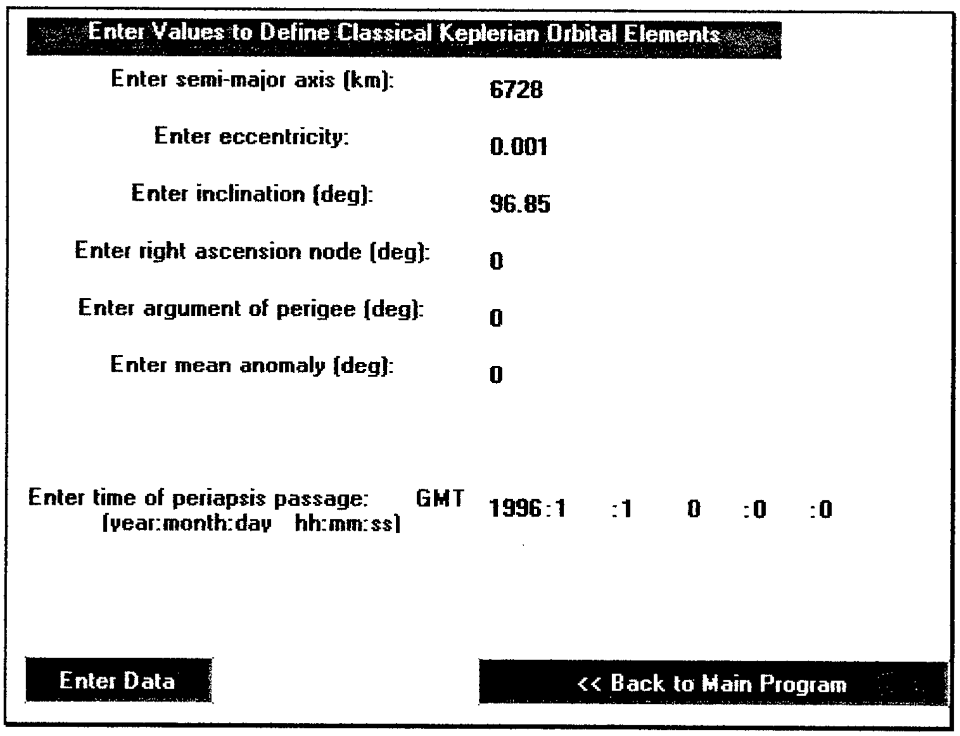


Figure 2-2: Screen for Orbital Parameters

Once all values are defined, depress <Enter Data> and a "done" statement will appear if successfully performed.

2.6 Part II: Build a Satellite

Building a satellite requires performing subsystem construction first as shown in Figure 2-3.

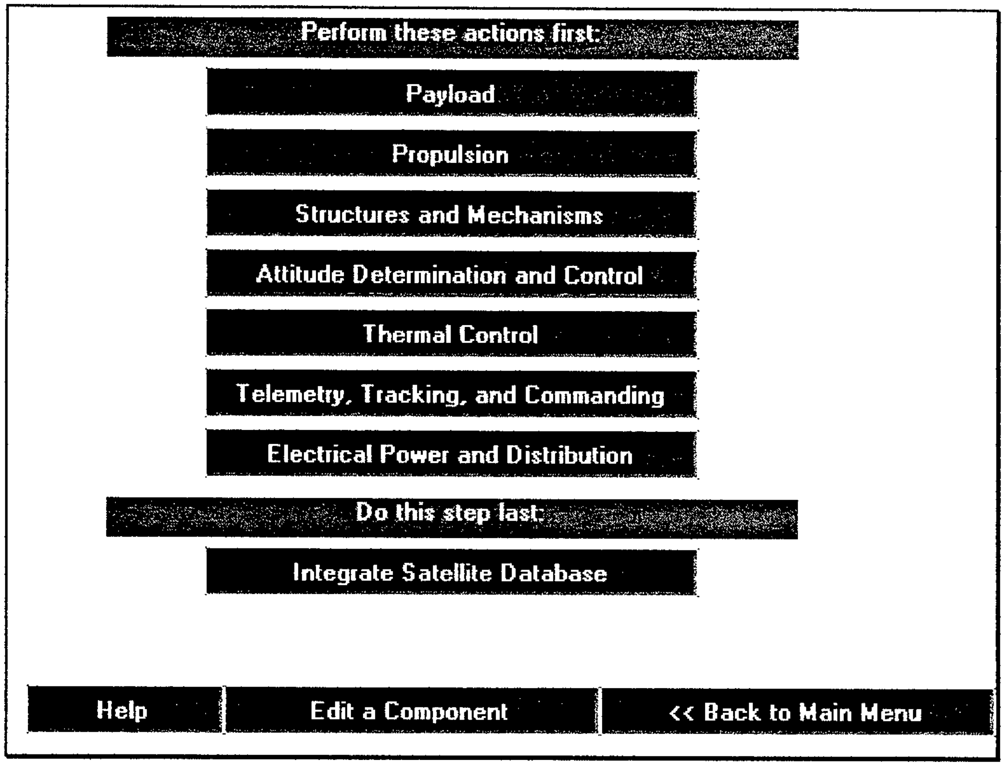


Figure 2-3: Screen for Building a Satellite

Once subsystems are completed, they may be integrated by pressing <Integrate Satellite Database>. For on line help, press <Help>. If a component needs editing, ensure the ModSat database is integrated first. If all subsystems are not complete, integration still occurs and an "incomplete" message is displayed. Components use ID numbers, so keep track of those numbers as subsystems are built for easy editing reference later.

2.6.1 Payload

Select the desired payload type from options as shown in Figure 2-4.

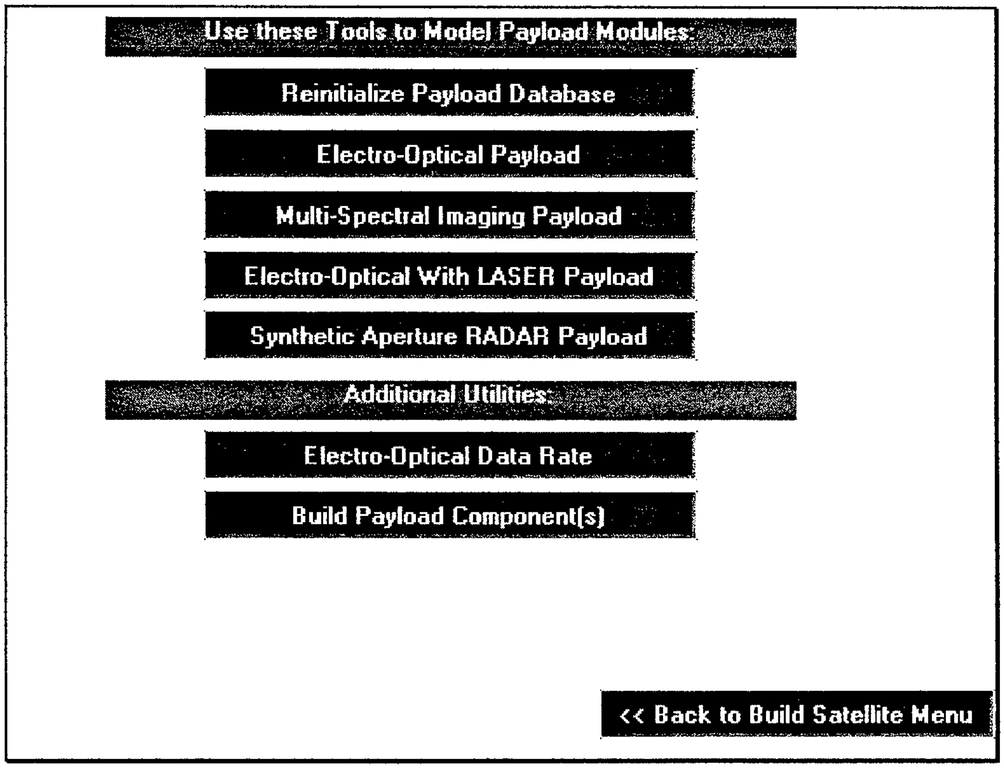


Figure 2-4: Payload Build Menu

Depress only one payload button, else for every payload button pushed, a new payload package will be added to ModSat's database as an intended mission module attached to the satellite! "Electro Optical Data Rate" provides calculations on how fast data is taken in for optical payloads, which is a driving factor for memory and processing requirements. A payload which is not in the main list can be custom built by using "Build Payload Component(s)."

2.6.2 Propulsion

Perform propulsion construction in the order as listed in Figure 2-5.

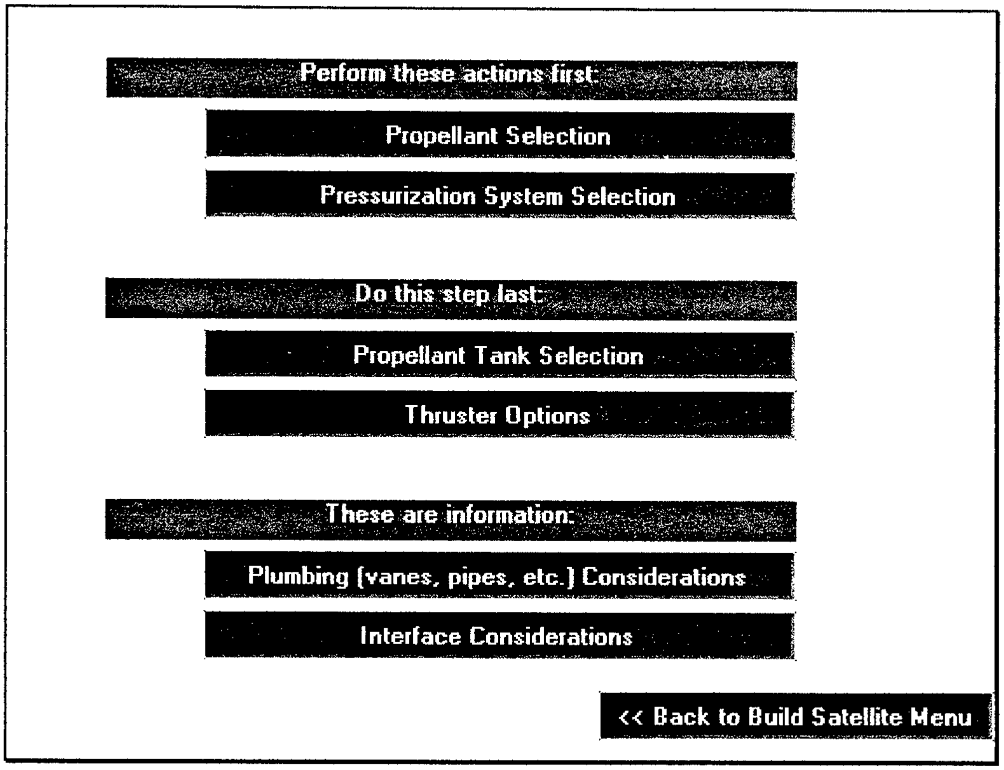


Figure 2-5: Propulsion Build Menu

Fuel may be mono- or bi-propellant.

2.6.3 Structures and Mechanisms

To create the physical satellite structure, multiple areas must be defined as shown in the opening screen in Figure 2-6:

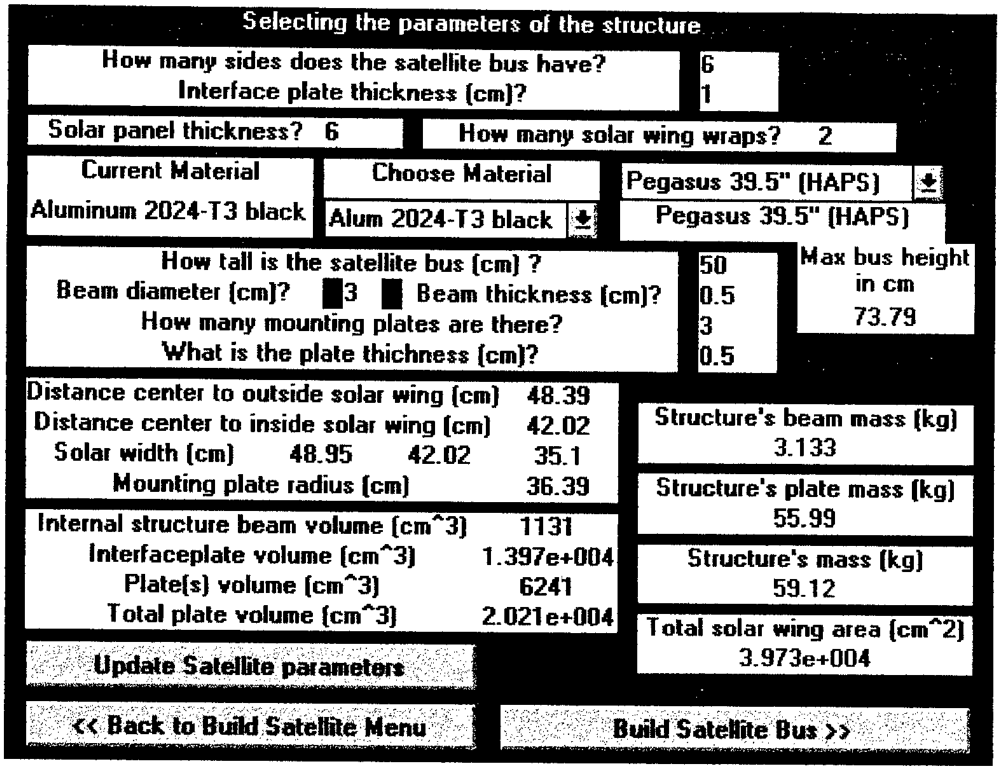


Figure 2-6: Structures and Mechanisms Build Menu

Once parameters are specified press <Update Satellite parameters>. If the update is satisfactory continue by pressing <Build Satellite Bus>.

2.6.4 Attitude Determination and Control System (ADCS)

Construct ADCS by performing the actions as listed in Figure 2-7.

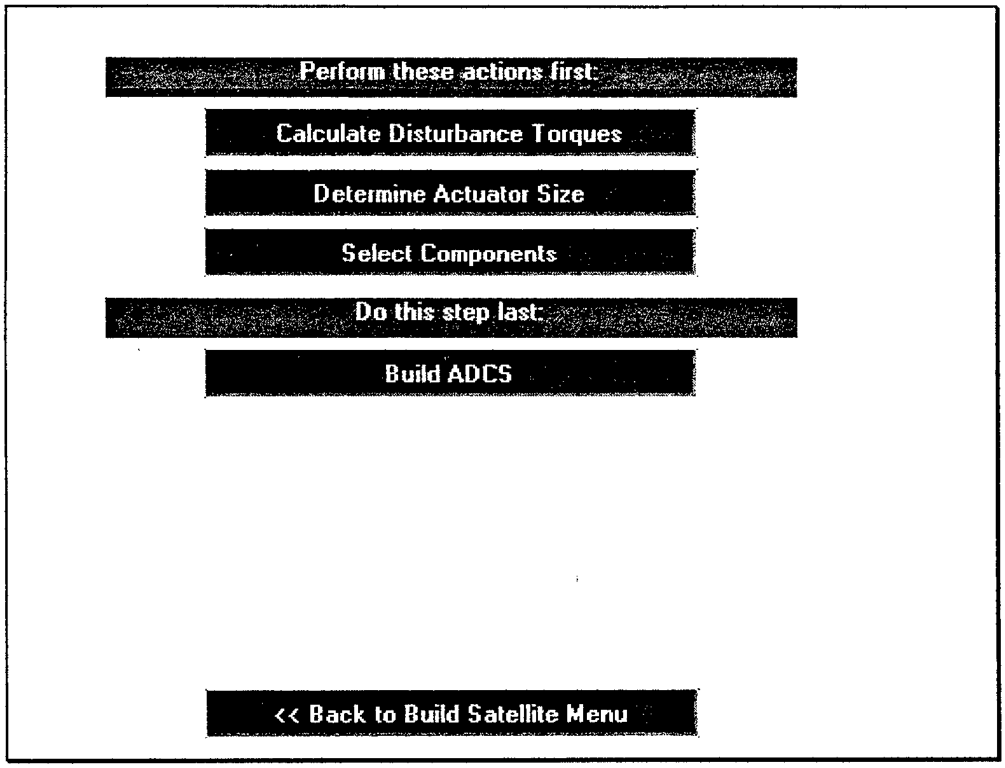


Figure 2-7: ADCS Build Menu

2.6.5 Thermal Control

If ModSat requires special thermal components, such as cryogenics for active cooling, these items may be constructed using the menu shown in Figure 2-8:

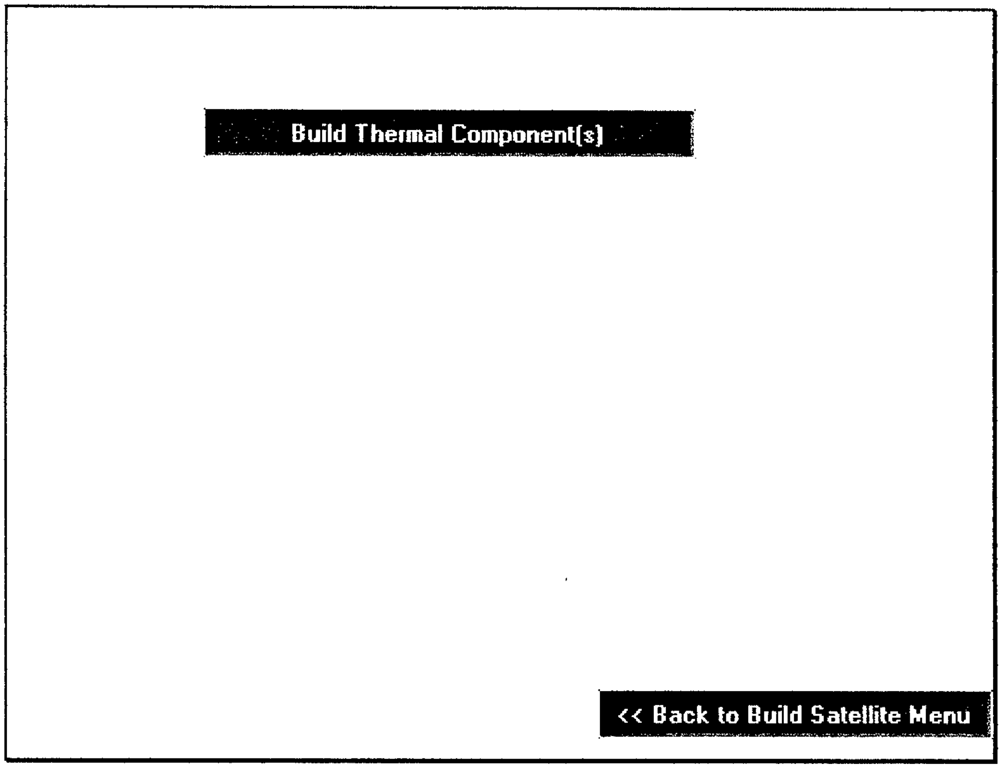


Figure 2-8: Thermal Build Menu

The built component(s) will be added to the ModSat database.

2.6.6 Telemetry, Tracking, and Commanding (TTC)

Perform TTC construction in the order listed as shown in Figure 2-9 for performing items first.

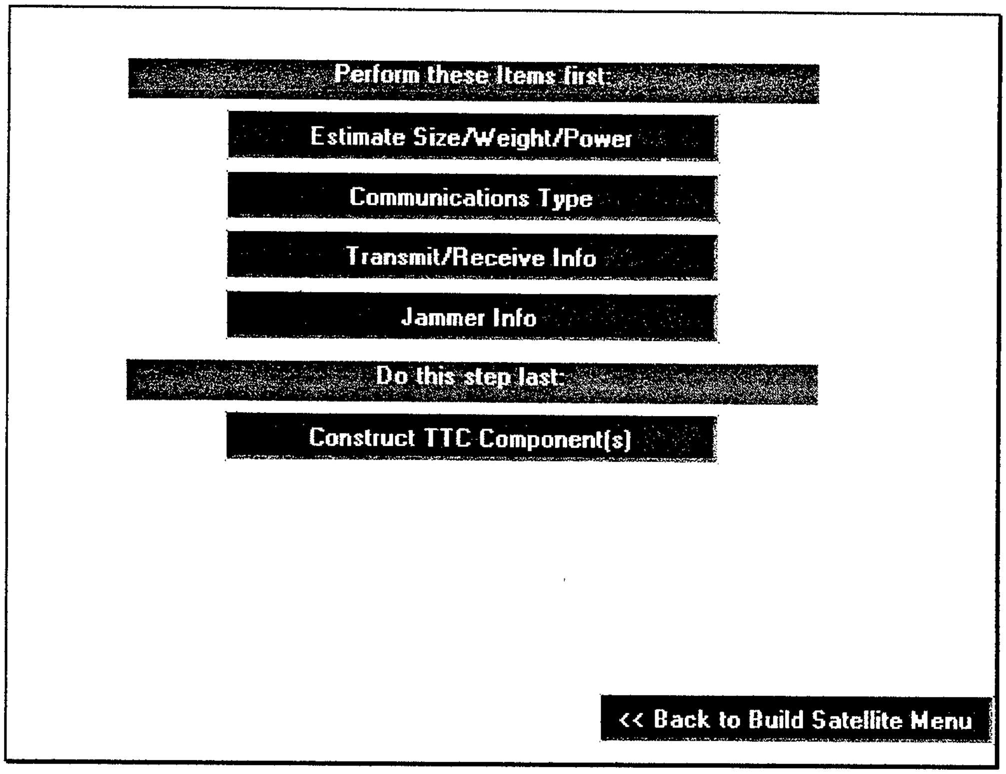


Figure 2-9: TTC Build Menu

"Estimate Size/Weight/Power" is provided in the event all data is not known, and estimates are necessary to help define component characteristics. One of the three characteristics can be known, with estimations given on the other two. This action is optional. "Communications Type" must be defined since the data is used to help calculate maximum data downlink rate supportable. "Transmit/Receive Info" is used to define communications links as pairs (i.e. transmitter to receiver). "Jammer Info" is optional and is used to define hostile action against ModSat. PCSoap uses this data to simulate jamming efforts against ModSat, which can give insight on how well the defined communications package works if threatened. Press <Construct TTC Component(s)> to build the physical TTC items onboard ModSat.

2.6.7 Electrical Power and Distribution

Perform actions in the order listed for the actions first list as shown in Figure 2-10, since calculations performed in one action feed into the next set of calculations.

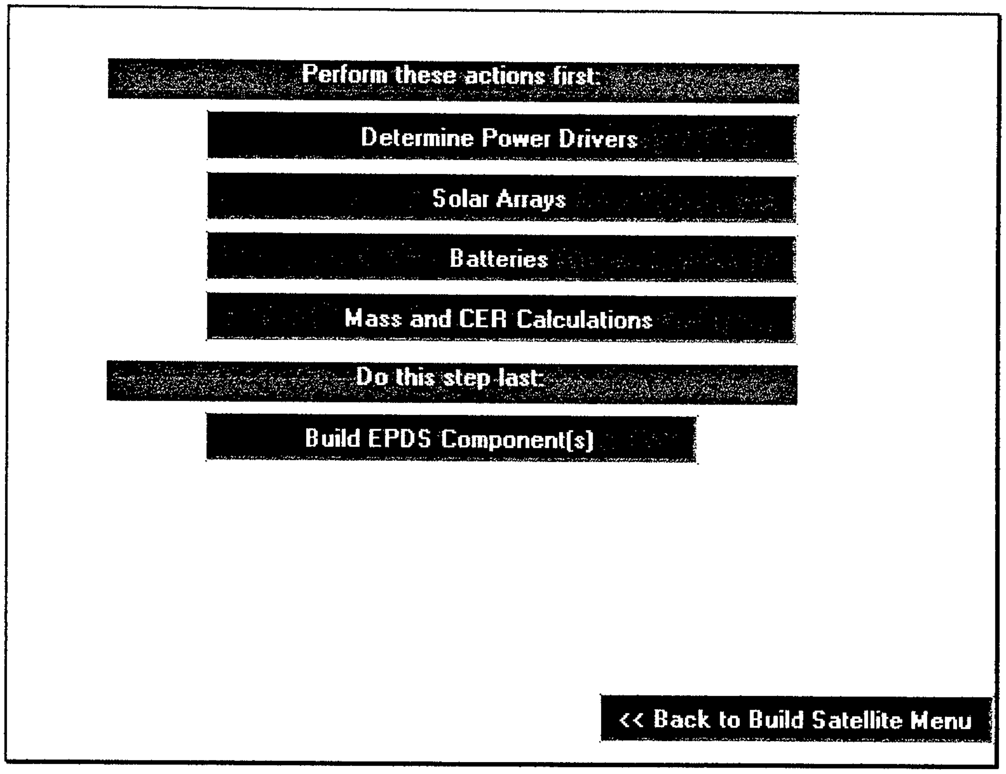


Figure 2-10: Electrical Build Menu

Once parameters are calculated, press <Build EPDS Component(s)> to build the physical electrical systems onboard ModSat.

2.7 Part III: Test a Satellite

Satellite tests include total spacecraft mass, subsystem component overlap check, and power constraints check as shown in Figure 2-11. The satellite can be displayed in 3D to get a better understanding of how ModSat is fitting (or not fitting) together.

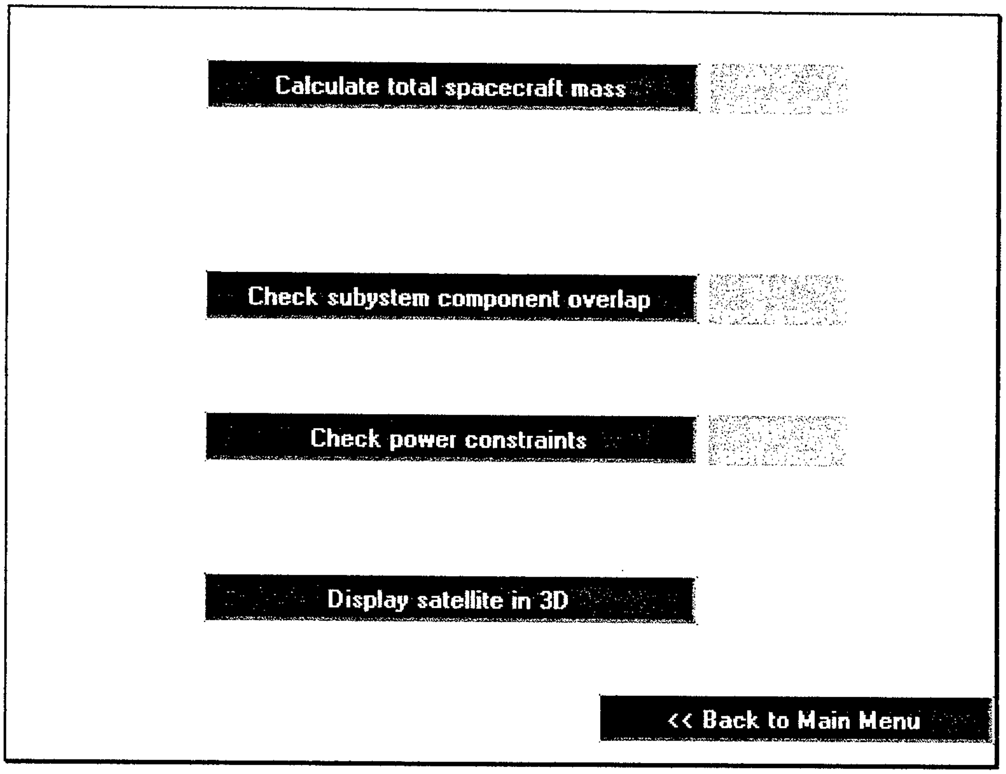


Figure 2-11: Test Satellite Menu

Buttons for calculating spacecraft center of gravity, total inertia matrix, and launch vehicle constraints appear on this screen as successful calculations are performed in the other tests.

2.7.1 Total Spacecraft Mass

The weight budget may be checked as the satellite is constructed as shown in Figure 2-12. Be sure the satellite database is integrated before accessing this feature.

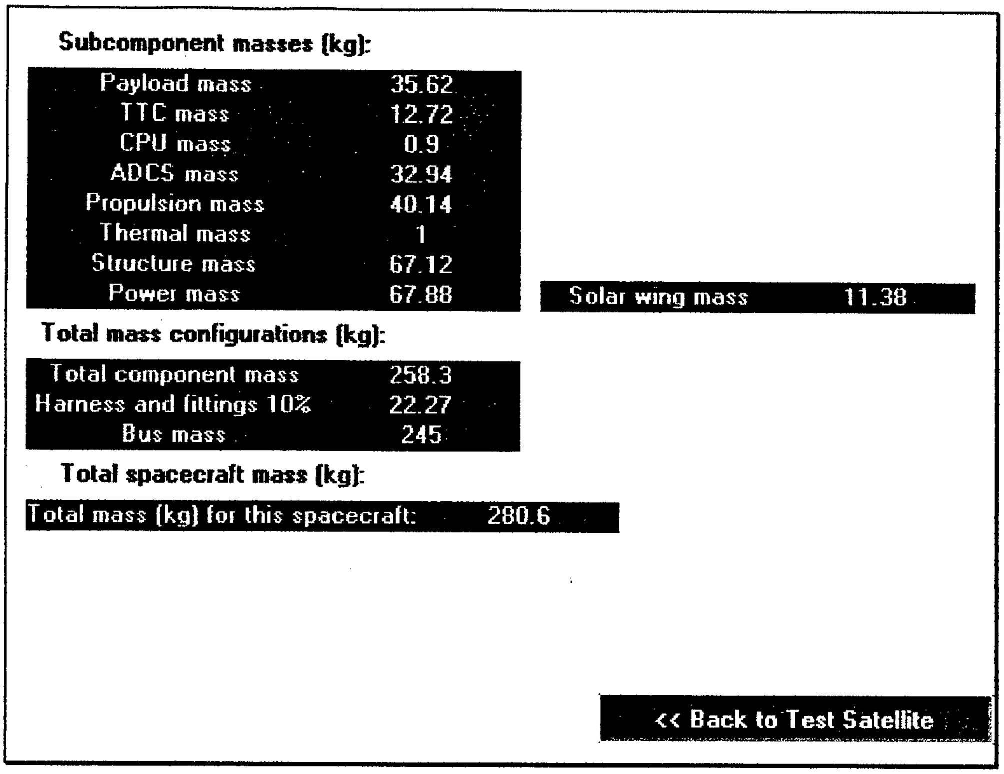


Figure 2-12: Total Mass Check

2.7.2 Subsystem Component Overlap

This function does not catch overlap conditions in all cases. The user is advised to check for overlap visually through use of the display satellite in 3D feature.

2.7.3 Power Constraints

The power budget may be checked as the satellite is constructed as shown in Figure 2-13. Be sure the satellite database is integrated before accessing this feature.

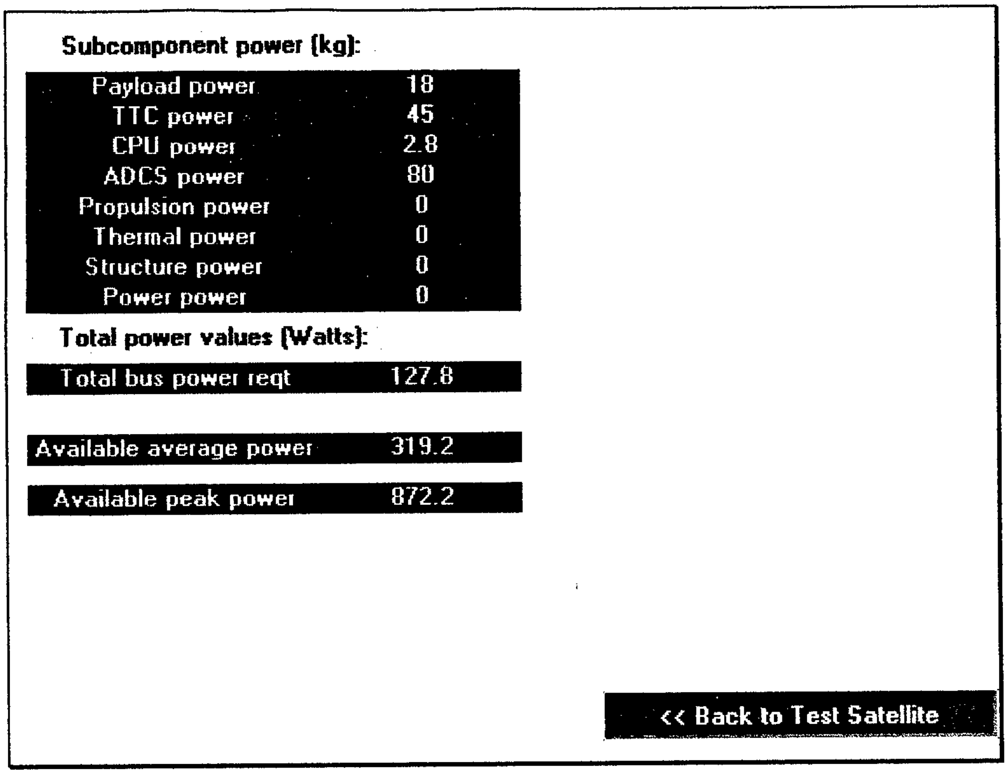


Figure 2-13: Power Constraint Check

2.7.4 Display Satellite in 3D

The satellite may be viewed by subsystem or in its entirety as shown in Figure 2-14, along with the capability to select axes values which gives the ability to zoom in or out. Viewing angle may be selected by use of slider bars while the satellite is displayed.

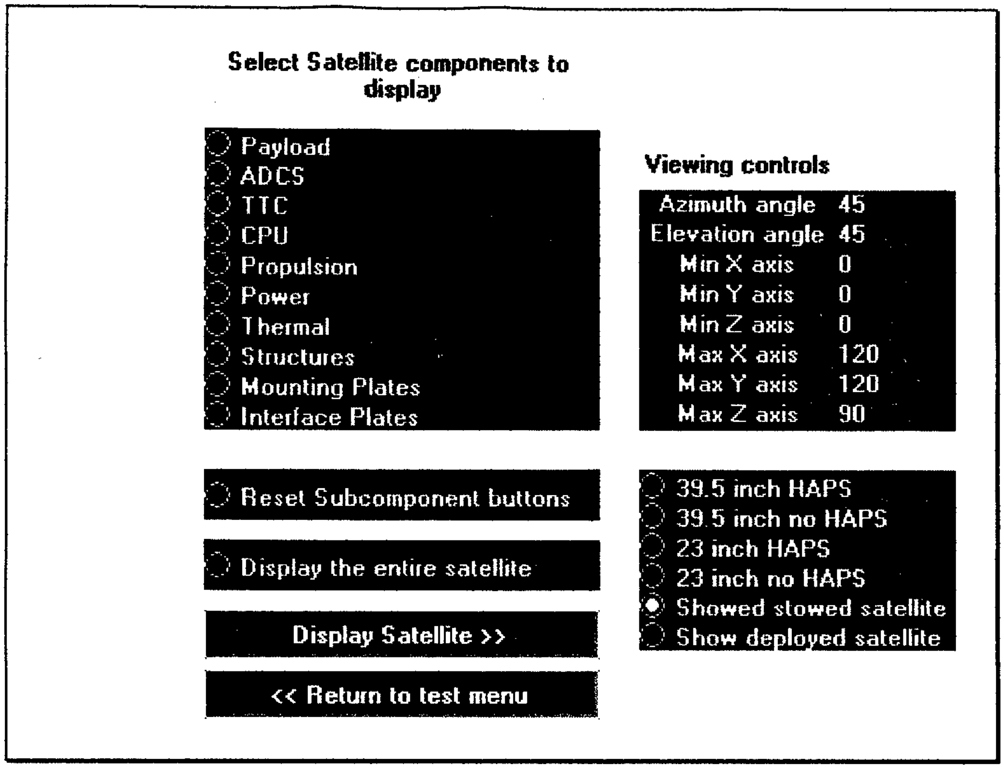


Figure 2-14: Menu Selection for Satellite Display in 3D

Choose the launch vehicle type and the 3D display will draw the appropriate payload faring around the satellite vehicle. This feature enables the user to check that satellite and its mission module fit within necessary volume restrictions; otherwise, protrusion will be obvious.

2.8 Part IV: Initialize Satellite

The satellite operating system is intended to simulate a self check upon power up to ensure all necessary subsystems are on-line, especially since the satellite is intended to be modular. If all necessary systems are present, the initialization screen resembles Figure 2-15.

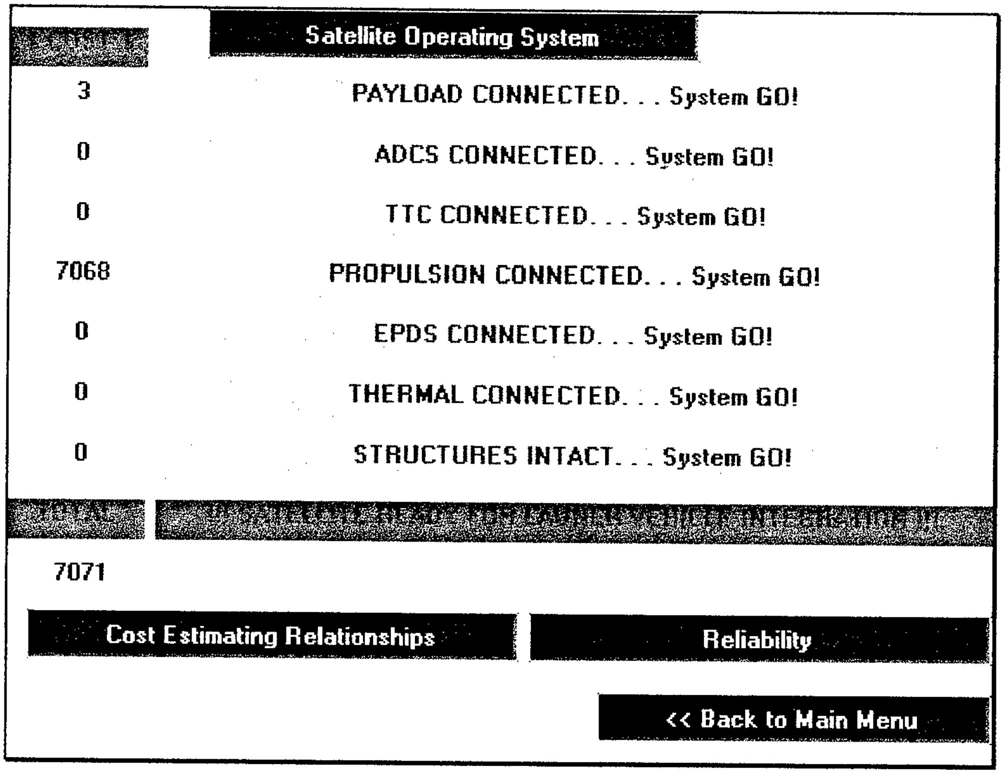


Figure 2-15: Initialize Satellite Screen

If systems are go, then buttons appear for accessing cost estimating relationships and reliability calculations. The cost column to the left is provided as a reference on how much subsystems cost if a user has entered cost data from some shopping list when building the subsystem components. Totals are given for each subsystem, but does not replace cost estimating relationship functions.

2.8.1 Cost Estimating Relationships (CER)

The CER menu as shown in Figure 2-16 provides the option of using one of two cost models as currently used by the Air Force.

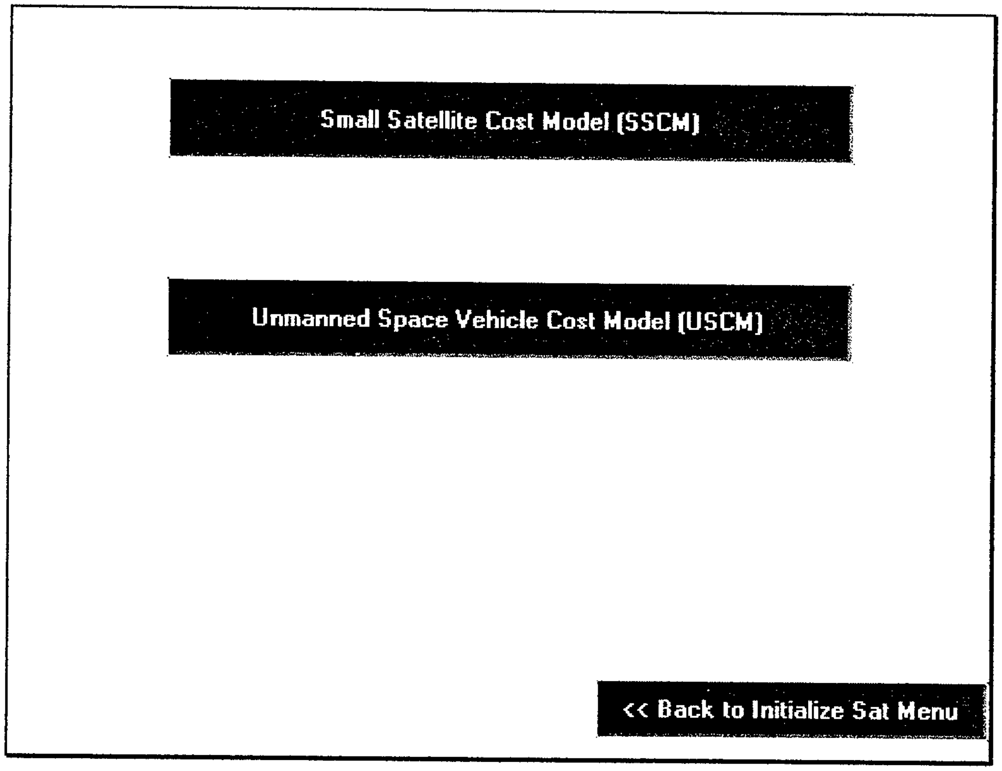


Figure 2-16: CER Selection Menu

Once a model is run, the values are used in the analysis section.

2.8.2 Reliability

Reliability calculations are provided as shown in Figure 2-17.

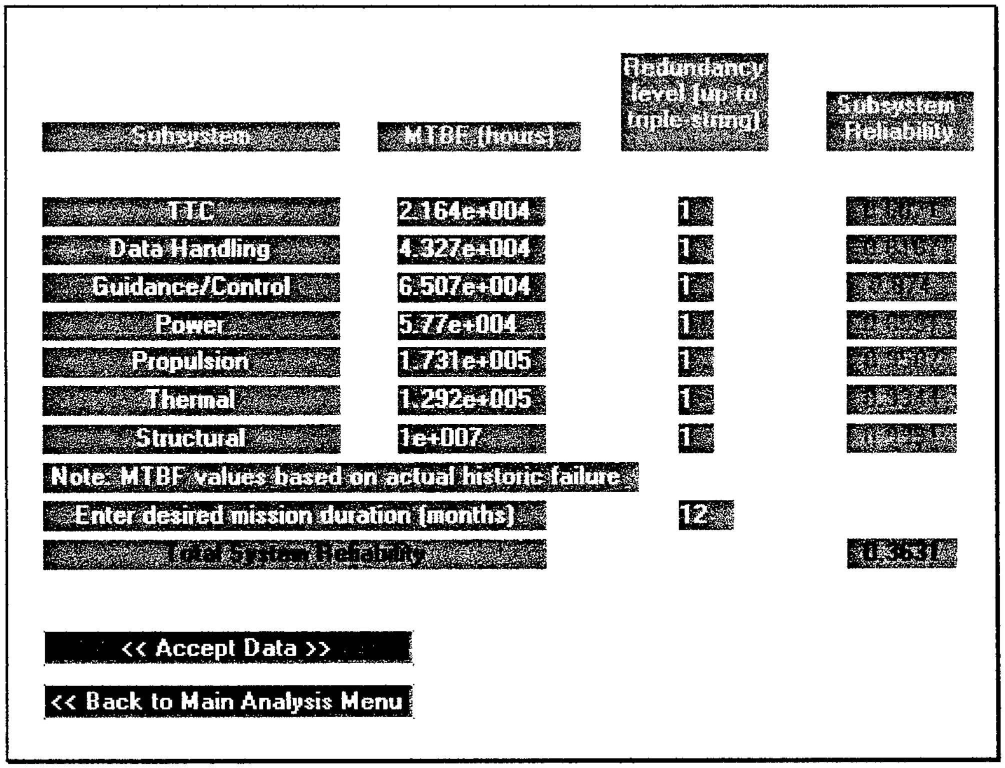


Figure 2-17: Reliability Calculation Screen

The user may define up to triple string systems, MTBF for each subsystem, and mission duration. Reliability data is used in the analysis section.

2.9 Part V: Run Scenarios

Once a satellite is defined and the database is integrated, the user may select from one of four scenarios as shown in Figure 2-18.

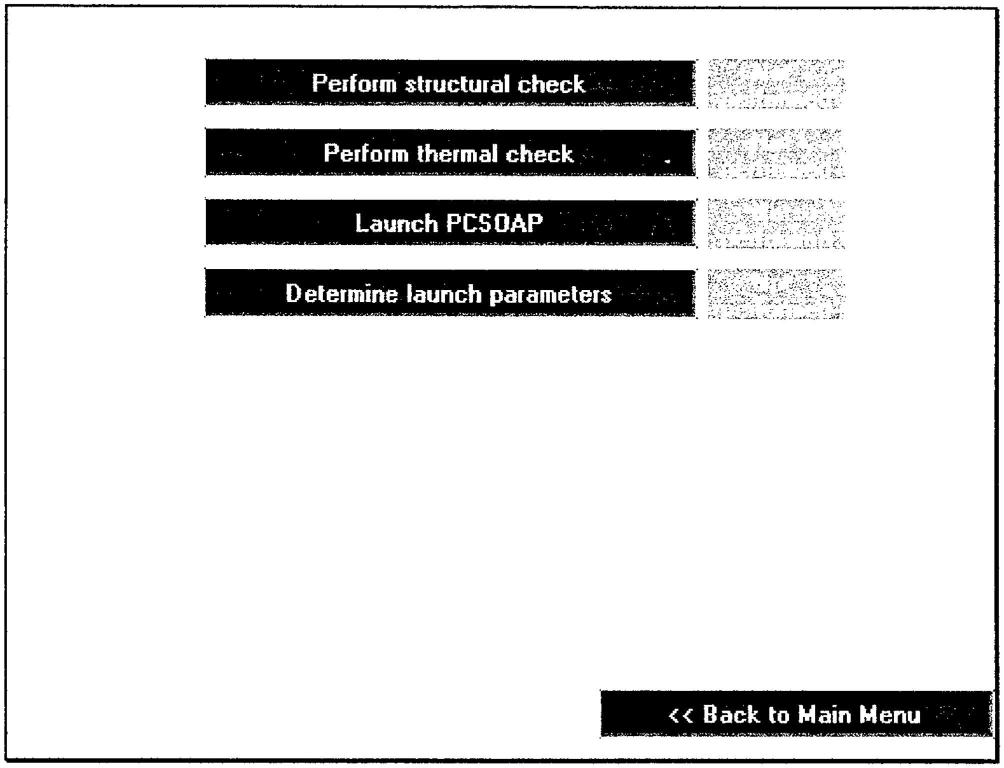


Figure 2-18: Run Scenarios Menu

The structural check gives feedback on whether or not the satellite is structurally sound enough to survive launch loads from the selected vehicle. Thermal check gives an indication of which components bust their tolerance to temperature extremes in the satellite's designated orbit. This information can be used to reposition components or provide thermal protection as necessary. Launch PCSoap creates the file necessary for PCSoap to read information and display ModSat in its designated orbit. Once the user is in this program any number of simulation analysis can be performed. Launch parameters provides compatibility feedback on the selected launch vehicle and ModSat.

2.10 Part VI: Analyze Data

After all tests have been run and the satellite has passed, the user can then use the analysis section. The main screen appears as in Figure 2-19.

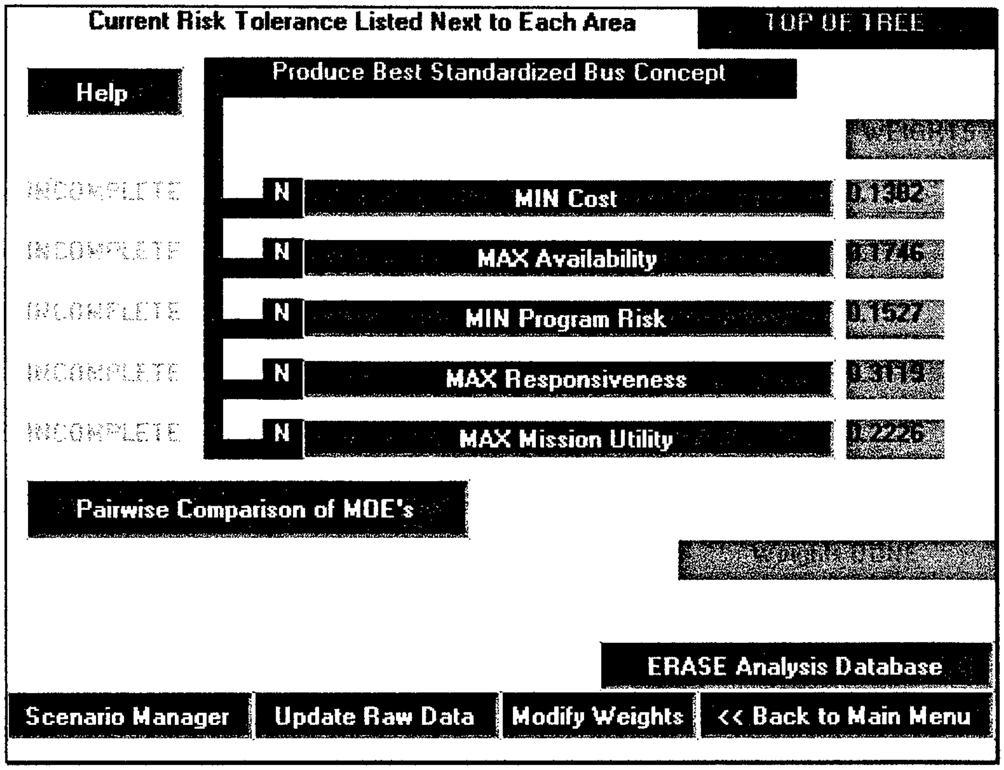


Figure 2-19: Data Analysis Screen

Pairwise comparison of Measures of Effectiveness (MOE's) enables the user to judge relative values of all 27 MOE's. This information is used to generate weights in each category, which is displayed to the right of the screen. Scenario manager enables the user to preload risk preferences, view an existing analysis, perform sensitivity analysis, or reperform an existing analysis. Update raw data must be used to extract all necessary information for the analysis. If all information is available a "done" message will appear; otherwise, the appropriate error message will flag a missing variable. Weights can be modified using the "modify weights" button, but all weights must sum to one. When a user wishes to start afresh, a button is provided to erase the existing database information. In the event this button is depressed by accident there is no problem since a warning message is displayed along with the opportunity to back out with data still intact.

At the MOE level the user may refine risk attitude or define a utility function. Once data is selected and entered the user must not return to that MOE for that satellite design since data is written to a database upon exiting a MOE screen. By returning to the same MOE screen after it is "done" the user causes ModSat to add information on top of existing information. This action can lead to erroneous results when analysis is performed.

Only when all 27 MOE screens have been accessed and data defined will the "analyze data" button appear on the main screen. When this action occurs the user may add the satellite information to an existing database, create a new one, or perform analysis. A feature is provided to enter a text description of the design. If only one design exists, ModSat will refuse to perform an analysis.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	November 1996	Master's Thesis

4. TITLE AND SUBTITLE

The Preliminary Design of a Standardized Spacecraft Bus for Small Tactical Satellites (Volume 3)

5. FUNDING NUMBERS

6. AUTHOR(S)

Gerald F. Ashby, Capt, USAF, et al.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology, WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GSE/GSO/ENY/96D-1

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Lt Col James Rooney
Phillips Laboratory/WSM
3550 Aberdeen Ave SE
KAFB, NM 87117-5776

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

A

13. ABSTRACT *(Maximum 200 words)*

Current satellite design philosophies concentrate on optimizing and tailoring a particular satellite bus to a specific payload or mission. Today's satellites take a long time to build, checkout, and launch. An alternate approach shifts the design paradigm to one that focuses on access to space, enabling tactical deployment on demand and the capability to put current payload technology into orbit, versus several years by today's standards, by which time the technology is already obsolete. This design study applied systems engineering methods to create a satellite bus architecture that can accommodate a range of remote sensing mission modules. System-level and subsystem-level tradeoffs provided standard components and satellite structures, and an iterative design approach provided candidate designs constructed with those components. A cost and reliability trade study provided initial estimates for satellite performance. Modeling and analysis based upon the Sponsor's objectives converged the designs to an optimum solution. Major products of this study include not only a preliminary satellite design to meet the sponsor's needs, but also a software modeling and analysis tool for satellite design, integration, and test. Finally, the report provides an initial implementation scheme and concept for operations for the tactical support of this satellite system.

14. SUBJECT TERMS

Satellite Bus, Spacecraft Design, Systems Engineering, Modular Satellite, Satellite Subsystems, Remote Sensing

15. NUMBER OF PAGES

94

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to *stay within the lines* to meet *optical scanning requirements*.

Block 1. Agency Use Only (*Leave blank*).

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C	- Contract	PR	- Project
G	- Grant	TA	- Task
PE	- Program Element	WU	- Work Unit Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. *(If known)*

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

- DOD - See DoDD 5230.24, "Distribution Statements on Technical Documents."
- DOE - See authorities.
- NASA - See Handbook NHB 2200.2.
- NTIS - Leave blank.

Block 12b. Distribution Code.

- DOD - Leave blank.
- DOE - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.
- NASA - Leave blank.
- NTIS - Leave blank.

Block 13. Abstract. Include a brief *(Maximum 200 words)* factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code *(NTIS only)*.

Blocks 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.